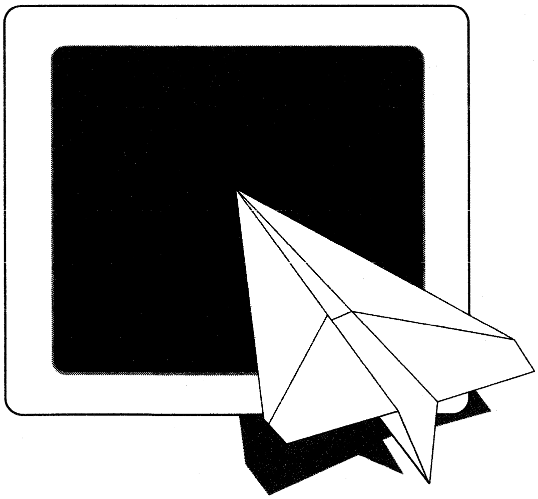


XACT Reference Guide, Volume 3

XACT Reference Guide, Volume 3 — April, 1994 (0401131 02)

Printed in U.S.A.

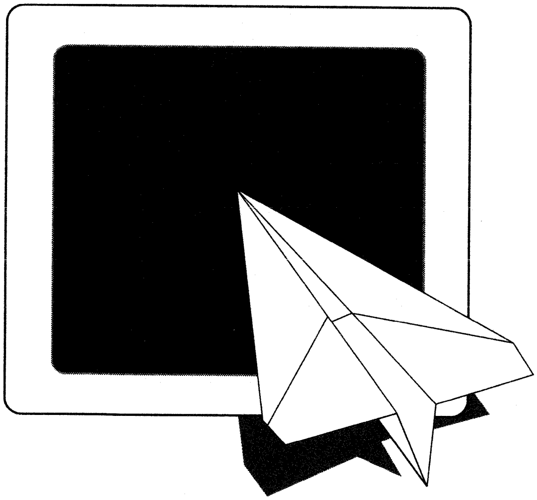
XILINX®, XACT, XC2064, XC3090, XC4005, and XC-DS501 are registered trademarks of Xilinx. All XC-prefix product designations, XACT-Performance, XAPP, X-BLOX, XChecker, XDM, XDS, XEPLD, XFT, XPP, XSI, BITA, Configurable Logic Cell, CLC, Dual Block, FastCLK, HardWire, LCA, Logic Cell, LogicProfessor, MicroVia, PLUSASM, UIM, VectorMaze, and ZERO+ are trademarks of Xilinx. The Programmable Logic Company and The Programmable Gate Array Company are service marks of Xilinx.

IBM is a registered trademark and PC/AT, PC/XT, PS/2 and Micro Channel are trademarks of International Business Machines Corporation. DASH, Data I/O and FutureNet are registered trademarks and ABEL, ABEL-HDL and ABEL-PLA are trademarks of Data I/O Corporation. SimuCad and Silos are registered trademarks and P-Silos and P/C-Silos are trademarks of SimuCad Corporation. Microsoft is a registered trademark and MS-DOS is a trademark of Microsoft Corporation. Centronics is a registered trademark of Centronics Data Computer Corporation. PAL and PALASM are registered trademarks of Advanced Micro Devices, Inc. UNIX is a trademark of AT&T Technologies, Inc. CUPL is a trademark of Logical Devices, Inc. Apollo and AEGIS are registered trademarks of Hewlett-Packard Corporation. Mentor and IDEA are registered trademarks and NETED, Design Architect, QuickSim, QuickSim II, and EXPAND are trademarks of Mentor Graphics, Inc. Sun is a registered trademark of Sun Microsystems, Inc. SCHEMA II+ and SCHEMA III are trademarks of Omation Corporation. OrCAD is a registered trademark of OrCAD Systems Corporation. Viewlogic, Viewsim, and Viewdraw are registered trademarks of Viewlogic Systems, Inc. CASE Technology is a trademark of CASE Technology, a division of the Teradyne Electronic Design Automation Group. DECstation is a trademark of Digital Equipment Corporation. Synopsys is a registered trademark of Synopsys, Inc. Verilog is a registered trademark of Cadence Design Systems, Inc.

Xilinx does not assume any liability arising out of the application or use of any product described herein; nor does it convey any license under its patents, copyrights, or maskwork rights or any rights of others. Xilinx, Inc. reserves the right to make changes, at any time, in order to improve reliability, function or design and to supply the best product possible. Xilinx, Inc. cannot assume responsibility for the use of any circuitry described herein other than circuitry entirely embodied in its products. Xilinx products are protected under at least the following U.S. patent: 5,224,056. Xilinx, Inc. does not represent that Xilinx products are free from patent infringement or from any other third-party right. Xilinx assumes no obligation to correct any errors contained herein or to advise any user of this text of any correction if such be made. Xilinx will not be liable for the accuracy or correctness of any engineering or software or assistance provided to a user.

Xilinx products are not intended for use in life support appliances, devices, or systems. Use of a Xilinx product in such applications without the written consent of the appropriate Xilinx officer is prohibited.

Table of Contents



Overview

XACT Reference Guide, Volume 3

XACT Reference Guide, Volume 3 — April, 1994 (0401131 02)

Printed in U.S.A.

XACT Reference Guide Overview

The *XACT Reference Guide* contains information on the software programs and hardware peripherals found in the XACT Development System. This guide is arranged into three volumes: Volume 1, “Design Entry and Conversion”; Volume 2, “Design Implementation”; and Volume 3, “Design Verification.” Generally, the chapters in these volumes are organized in the following manner.

- A brief summary of program functions
- A syntax statement
- A review of the input files used and the output files generated by the program
- A listing of the commands, options or parameters used by the program
- Examples of how you can use the program
- Warning and error messages provided by the program with suggested recovery procedures for the error messages when appropriate

Typographical Conventions

The following conventions are used in this manual’s syntactical statements:

`Courier font regular`
System messages or program files appear in regular Courier font.

`Courier font bold`
Literal commands that you must enter in syntax statements are in bold Courier font.

italic font
Variables that you replace in syntax statements are in italic font.

[ ]
Square brackets denote optional items or parameters. However, in bus specifications, such as bus [7:0], they are required.

{ }
Braces enclose a list of items from which you must choose one or more.

.
.
.
A vertical ellipsis indicates material that has been omitted.

... A horizontal ellipsis indicates that the preceding can be repeated one or more times.

| A vertical bar separates items in a list of choices.

↵ This symbol denotes a carriage return.

The Design Flow

Figure 3-1 shows the three parts of the LCA™ design flow: design entry, design implementation, and design verification.

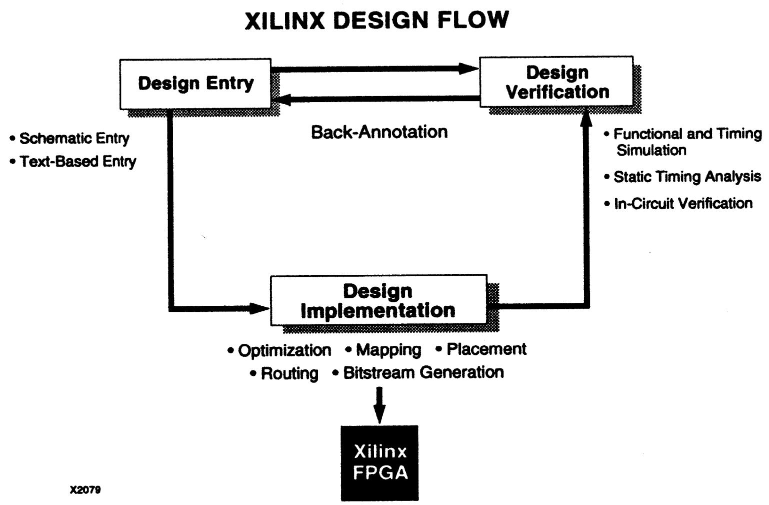


Figure 3-1 The XACT Development System Design Flow

Design entry is the process that takes the design from concept to netlist. There are a number of ways to enter a design, including schematics, Boolean or state expressions, and hardware description languages (HDLs). These entry methods require third-party tools that produce a design file in its own netlist format.

The process of design implementation converts the third-party netlist to Xilinx Netlist Format (XNF). It ultimately produces a configuration bitstream for the target LCA device. It includes optimization and mapping, placement and routing, and bitstream creation. Designs can be implemented automatically or by using a combination of the automatic and manual Xilinx Development System tools.

Design verification includes simulation, static-timing analysis, and in-circuit verification. Simulation is performed on third-party tools supported by Xilinx. The input for these tools requires a tool-specific translation of an XNF file. You can simulate an XNF file at any point or convert to an XNF file. Static-timing analysis tools and in-circuit verification tools are part of the Xilinx Development System. Consult the Design Verification chapter in the *XACT User Guide* for a more detailed description.

In some cases schematic simulation can be performed from the tool that created the design. For these tools, it is possible to simulate a design without creating an XNF file. As a result, simulator input can be another output of the design-entry tools.

Functional Sections of the *XACT Reference Guide*

The *XACT Reference Guide* provides detailed information about converting, implementing, and verifying designs in the XACT environment. Check the program chapters for information on what program works with each family of LCA device. The following is a brief overview of the contents and organization of the *XACT Reference Guide.*

Volume 1 — Design Entry and Conversion

Overview

The Overview provides generic descriptions of reference guide chapter structure, the typographical conventions used, a design flow summary, and descriptions of the programs in the XACT Development System.

XACT Design Manager (XDM)

XDM is the user interface shared by all XACT development programs.

XMake

XMake automatically converts a schematic design file into LCA and BIT files.

MemGen

MemGen creates RAMs or ROMs of any size for use in XC4000 designs.

XACT-Performance Utility

XACT-Performance™ specifies timing requirements on a schematic for any family (XC3000A/L, XC3100A, XC4000A/H) that PPR uses.

XNFCVT

XNFCVT converts an XNF netlist from Version 5 to Version 4, 2, or 1.

HM2RPM

HM2PRM converts hard macros to relationally placed macros (RPMs).

Volume 2 — Design Implementation

Overview

The Overview provides generic descriptions of reference guide chapter structure, the typographical conventions used, a design flow summary, and descriptions of the programs in the XACT Development System.

XNFMerge

XNFMerge merges XNF and MAP files together to form a single XNF file.

XNFPrep

XNFPrep performs a design rule check (DRC) and removes unused and redundant logic from a flattened XNF file. It also checks the syntax of the XACT-Performance parameters found in the design and prepares delay information for PPR path analysis.

XNFMAP

XNFMAP maps the logic defined by an XNF file into LCA elements such as CLBs, IOBs, and TBUFs.

MAP2LCA

MAP2LCA translates a MAP file into an LCA file for input into APR.

APR (Automatic Place and Route)

APR places and routes XC2000, XC2000L, XC3000, and XC3100 designs.

PPR (Partition, Place and Route)

PPR partitions, places, and routes XC3000A/L, XC3100A, XC4000, and XC4000A/H designs.

MakeBits

MakeBits creates a configuration bitstream for the LCA design.

MakePROM

MakePROM converts a configuration bitstream (BIT) file into a file that can be downloaded to a PROM. MakePROM also combines multiple BIT files for use in a daisy chain of LCA devices.

Volume 3 — Design Verification

Overview

The Overview provides generic descriptions of reference guide chapter structure, the typographical conventions used, a design flow summary, and descriptions of the programs in the XACT Development System.

XDelay

XDelay is a static-timing analyzer that reports detailed timing information about the design and can be used for overall performance analysis.

LCA2XNF

LCA2XNF converts an LCA file to a Xilinx Netlist Format (XNF) file for use in timing simulation.

XNFBA

XNFBA creates an XNF file containing delay information and using the original schematic net names.

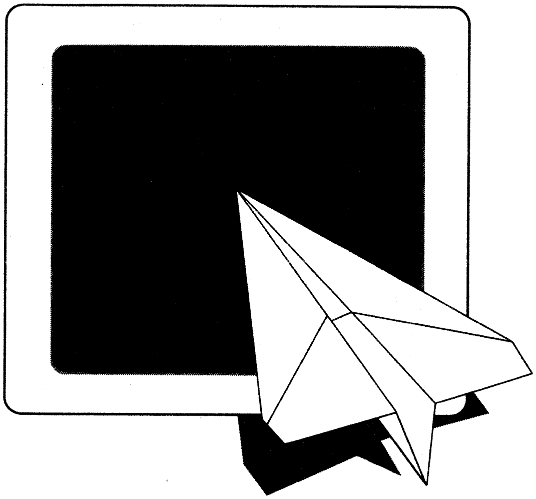
XACT Design Editor (XDE)

XDE allows the designer to manually edit or analyze an LCA design. XDE can also invoke MakeBits and MakePROM.

Other Xilinx Manuals

Other Xilinx manuals that you can refer to for more information are the following:

- *XACT User Guide*
- *XACT Reference Guide*, 3-Volume Set
- *XACT Libraries Guide*
- *Xilinx Hardware & Peripherals Guide*
- *XEPLD Design Guide*
- *XEPLD Reference Guide*
- *Xilinx-Synopsys Interface User Guide*
- *Xilinx ABEL Interface User Guide*
- *Viewlogic Interface User Guide*
- *OrCAD Interface User Guide*
- *Mentor V8 Interface User Guide*
- *The Programmable Logic Data Book*



The XDelay Timing Analysis Program

XACT Reference Guide, Volume 3

XACT Reference Guide, Volume 3 — April, 1994 (0401131 02)

Printed in U.S.A.

The XDelay Timing Analysis Program

This Program is Compatible with the Families Indicated.

☑ XC2000
☑ XC2000L
☑ XC3000
☑ XC3100
☑ XC3000A
☑ XC3000L
☑ XC3100A
☑ XC4000
☑ XC4000A
☑ XC4000H
□ XC7200
□ XC7300

The XDelay program is a static timing analysis tool that reports the timing delays of a routed LCA design. XDelay can create a detailed list of delay paths, and provide a design analysis, including estimates of system performance for each clock in the design.

XDelay has two modes: interactive and batch. The interactive mode is available within the XACT environment and the operating system prompt. The batch mode is only available from the operating system prompt. The graphical interface provides you with interactive menu access to the commands and options; the batch mode provides you text-based command syntax interface.

Graphical Interface

The graphical interface provides a menu-driven environment for you to use XDelay. This interface consists of pull-down menus that contain commands and options to specify design information used in timing analysis.

- You can start XDelay from the operating system prompt.
- You can use XDelay from the XACT environment by selecting XDE (XACT Design Editor), starting the EditLCA program, and selecting XDelay from the Timing menu.

Starting XDelay from the Operating System

You can call XDelay by typing the program name only at the operating system prompt.

```
xdelay
```

NOTE

If you type any options after the command name, you start the XDelay program in batch mode. Details about this interface appear later in this chapter.

The "Graphical Interface" section describes the menus and commands that are available in the graphical user interface.

Starting XDelay from within XACT

You can use XDelay from within the XACT environment by first starting XDM (XACT Design Manager). With the XDM screen open you then have two ways to select and run XDelay.

One way is start XDelay directly. To do this, place the mouse pointer on the Verify menu and select XDelay (Static Delay Analyzer). A list of XDelay command options appear on a pull-down menu from which you make selections. If you select any of these options, XDelay starts up in the batch (non-interactive) mode. For information about these options, refer to the Text Interface section, later in this chapter.

You can also run the XDelay command from within XDE. To do this, you must start XDM. When the XDM screen appears, select XDE from the PlaceRoute menu. When the XDE screen appears, select EditLCA from the Programs menu. EditLCA is the graphical editor for your LCA design. Once EditLCA has started, select XDelay options from the Timing menu.

Graphical Interface Commands

This section describes the commands available in the XDelay graphical interface. The XDelay executive manager graphical interface has four menus (Design, Timing, Misc, and Profile) and a command line (on PCs, at the bottom of the screen; on workstations, at the top of the screen).

You can use a menu selection or command-line entry to run XDelay commands. If a submenu appears, you can directly select from the submenu or type commands from the keyboard. Each menu item entry highlights the valid keyboard shortcut for entering the command or option on the command line either with a different color on color monitors, or capital letters.

The Design Menu

The Design menu contains commands that allow you to specify your working directory, and select a design from your working directory.

Directory

The Directory command allows you to change the current working directory from a list of available directories. You choose a design to analyze from the current working directory.

Design

The Design command allows you to select your design from the designs in the current directory. After you choose the design, XDelay loads it into memory.

Timing Menu

The Timing menu contains commands that allow you to choose three different modes of reporting timing delay information. If you want detailed information about specific paths in the design, use the Xdelay command. If you want to compare the implementation of the design against the original XACT–Performance TimeSpec constraints, use the Xdelay–TimeSpec command. If you want a summary of the overall performance of the design, use the Analyze command. In addition, the Speed command allows you to change the current speed grade for the design.

The Timing menu also contains several commands that allow you to:

- Create, delete, and list elements of specified TimeGroups
- Set, delete, and list currently defined Time Specifications
- Save, read, clear, and list current template files
- Set, clear, save, and read currently defined margin (timing delays)
- Enable or disable report information about special block delays
- Report information about nets in the current design
- Run DRC (Design Rules Checker) on the current design.

XDelay

XDelay reports detailed timing information about your design. The report contains the worst-case path delays for the set of selected paths.

You can specify a path in one of two ways.

- Select starting points and ending points (to and from)
- Specify a path type to follow

The starting point or path source can be any point where a timing path can begin, typically an input IOB, or the output of a flip-flop or latch. The ending point or path load (path destination) can be any point where a timing path ends, typically an ouput IOB, or the input of a flip-flop or latch.

For example, using starting and ending points, you can trace every input pad that eventually connects to a flip-flop. You can specify a certain path type by tracing all flip-flop outputs that connect to flip-flop inputs.

XDelay–TimeSpec

XDelay–TimeSpec compares the implementation of the design with the original XACT–Performance TimeSpec constraints that were defined by the user. The report generated by this command summarizes the paths for each selected TimeSpec and indicates whether the paths passed or failed the timing specification limit.

The following table summarizes the commands available in the sub–menus for the XDelay and XDelay–TimeSpec Commands. These options are described in detail later in this document.

<table>
<tr><th>XDelay Sub-Menu Options</th><th>XDelay-TimeSpec Sub-Menu Options</th></tr>
<tr><td>–From</td><td>–SelectSpec</td></tr>
<tr><td>–FromIOB</td><td>–FailedSpec</td></tr>
<tr><td>–FromFF</td><td>–Unspecified</td></tr>
<tr><td>–FromAll</td><td>–TSMaxpaths</td></tr>
<tr><td>–To</td><td>–NoTimeGroups</td></tr>
<tr><td>–ToIOB</td><td></td></tr>
<tr><td>–ToFF</td><td></td></tr>
<tr><td>–ToAll</td><td></td></tr>
<tr><td>–PadToPad</td><td></td></tr>
<tr><td>–PadToSetup</td><td></td></tr>
<tr><td>–ClockToPad</td><td></td></tr>
<tr><td>–ClockToSetup</td><td></td></tr>
<tr><td>–ClockInput</td><td></td></tr>
<tr><td>–SourceClock</td><td></td></tr>
<tr><td>–NoSourceClock</td><td></td></tr>
<tr><td>–DestClock</td><td></td></tr>
<tr><td>–NoDestClock</td><td></td></tr>
<tr><td>–BreakLoop</td><td></td></tr>
<tr><td>–NoBreakLoop</td><td></td></tr>
<tr><th colspan="2">Common Options in XDelay and XDelay-TimeSpec Sub-Menus</th></tr>
<tr><td colspan="2">–WorstCase</td></tr>
<tr><td colspan="2">–IgnoreNet</td></tr>
<tr><td colspan="2">–NoIgnoreNet</td></tr>
<tr><td colspan="2">–Netfilter</td></tr>
<tr><td colspan="2">–NoNetfilter</td></tr>
<tr><td colspan="2">–Delayless</td></tr>
<tr><td colspan="2">–Delaygreater</td></tr>
<tr><td colspan="2">–Widereport</td></tr>
<tr><td colspan="2">–Shortreport</td></tr>
<tr><td colspan="2">–Sort</td></tr>
<tr><td colspan="2">–Maxpaths</td></tr>
<tr><td colspan="2">–ClearOptions</td></tr>
</table>

XDelay and XDelay–TimeSpec Command Options

The following list describes the options within the XDelay and XDelay–TimeSpec commands. You select an option by typing that option on the command line or selecting with the mouse from the menu.

–From

–From allows you to specify starting points for the path-delay calculator. Starting points can be pins, blocks, nets, or a wildcard (*) to represent all sources.

–FromIOB

–FromIOB allows you to select IOBs as starting points for the path-delay calculator from a menu of valid IOBs. This option requires that the BlkMenus option be enabled if you have invoked XDelay from the XACT Design Editor.

–FromFF

–FromFF allows you to select flip-flop nets as starting points for the path-delay calculator from a menu of valid flip-flop nets. Flip-flop nets are named by their output net names. You can select the source net from this menu, or enter it from the keyboard. This option requires that the NetMenus option be enabled if you have invoked XDelay from the XACT Design Editor.

–FromAll

–FromAll allows you to select paths to begin at any path source.

–To

–To allows you to select the ending points for the path-delay calculator. Ending points can be pins, blocks, nets, or a wild card (*) to represent all destinations.

–ToIOB

–ToIOB allows you to select IOBs as path loads ending points for the path-delay calculator from a menu of valid IOBs. This option requires that the BlkMenus option be enabled if you have invoked XDelay from the XACT Design Editor.

–ToFF

–ToFF allows you to select flip-flop nets as ending points from a menu of valid flip-flop output nets.

–ToAll

–ToAll allows you to select paths to end at any path destination. This option is the default.

–SelectSpec

–SelectSpec allows you to select user-defined TimeSpecs (timing specifications) to analyze. This option causes the report to summarize the paths for each selected timespec and indicate whether the paths passed or failed the timing specification limit.

–FailedSpec

–FailSpec allows you to restrict XDelay to only print the paths that do not meet your timing specifications.

–Unspecified

–Unspecified allows you to include paths that are not specified by any TimeSpec in the XDelay report. This option has no effect if all timing paths have a TimeSpec.

–NoTimeGroups

–NoTimeGroups prevents XDelay from listing the members of each TimeGroup at the beginning of a report. Since this information can be lengthly, the –NoTimeGroups option is on by default.

–SourceClock

–SourceClock allows you to set the selected clock net to be the starting point for all path searches. You can select multiple clock nets at the same time. This starts the path searches only at the flip-flops or latches that are clocked by the specified clock signal.

You must specify the active edge (rising, falling, or any) of the clock. Specifying the active edge limits the delay report to paths that start or end on rising, falling, or any clock edge.

–NoSourceClock

–NoSourceClock allows you to clear any previously defined source clocks.

–DestClock

–DestClock allows you to set the selected clock net to be the ending point for all path searches. You can select multiple clock nets at the same time. This option ends the path searches at all the flip-flops or latches that are clocked by the specified clock signal.

You must specify the active edge (rising, falling, or any) of the clock. Specifying the active edge limits the delay report to paths that start or end on rising, falling, or any clock edge.

–NoDestClock

–NoDestClock allows you to clear any previously defined destination clocks.

–Ignorenet

–Ignorenet allows you to specify which nets to ignore. You can use this option to specifically avoid tracking false paths through illegal-state reset logic.

–NoIgnorenet

–NoIgnorenet allows you to clear any previously defined Ignorenets.

–Netfilter

–Netfilter allows you to restrict the path tracer only to paths passing through netfilter nets. Paths that do not contain any netfilter nets are not reported.

–NoNetfilter

–NoNetfilter allows you clear any previously defined Netfilter nets. When you select this option, XDelay reports all traced paths, not just paths with previously specified filter nets.

NOTE

The SourceClock, DestClock, IgnoreNet and Netfilter options present menus of all valid nets only when NetMenus is enabled if you have invoked XDelay from the XACT Design Editor.

–BreakLoop

–BreakLoop allows you to mark nets as synchronous sources. XDelay treats these nets similarly to flip-flops. Timing paths can begin and end at the specified BreakLoop but not go through them. This allows output of latches built from combinational logic to be treated as synchronous timing elements. For more information refer to the discussion of BreakLoops later in this chapter.

This is similar to the Flagblk –Synchronous option, except that –BreakLoop provides a more resolute way to specify synchronous outputs.

–NoBreakLoop

–NoBreakLoop allows you to remove all existing BreakLoops.

NOTE *–PadToPad, –PadToSetup, –ClockToSetup, –ClockToPad, and –ClockInput are complementary, not mutually exclusive. For example, if you select both –PadToPad and –PadToSetup, XDelay reports on both types of paths.*

–PadToPad

–PadToPad allows you to restrict the report to paths that start at input pads and end at output pads. See Figure 3-2.

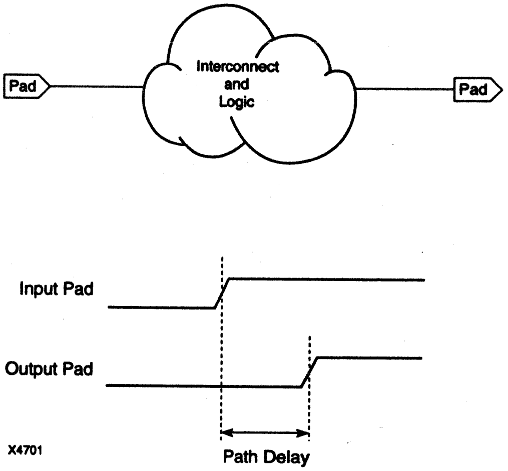


Figure 3-2 Pad To Pad Path

–PadToSetup

–PadToSetup allows you to restrict the report to paths that start at input pads and end at flip-flops or RAM. Figure 3-3 shows an example of a path from an input pad to a flip-flop data input .

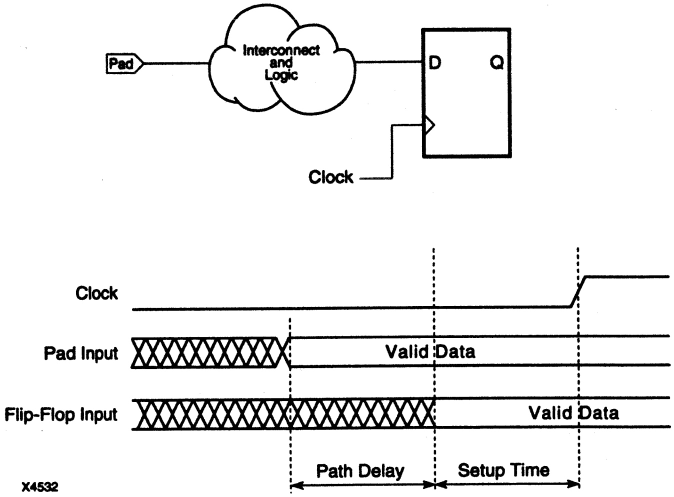


Figure 3-3 Pad To Setup Path

–ClockToPad

–ClockToPad allows you to restrict the report to paths that start at flip-flop outputs and end at output pads. See Figure 3-4.

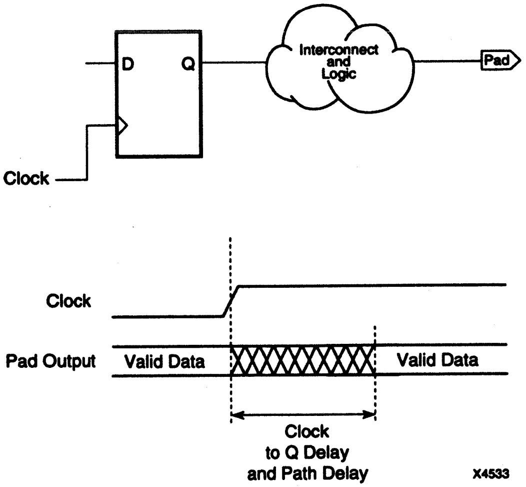


Figure 3-4 Clock To Pad Path

–ClockToSetup

–ClockToSetup allows you to restrict the report to paths that start at flip-flop outputs and end at clocked inputs, such as flip-flop data inputs. Refer to Figure 3-5 through Figure 3-8.

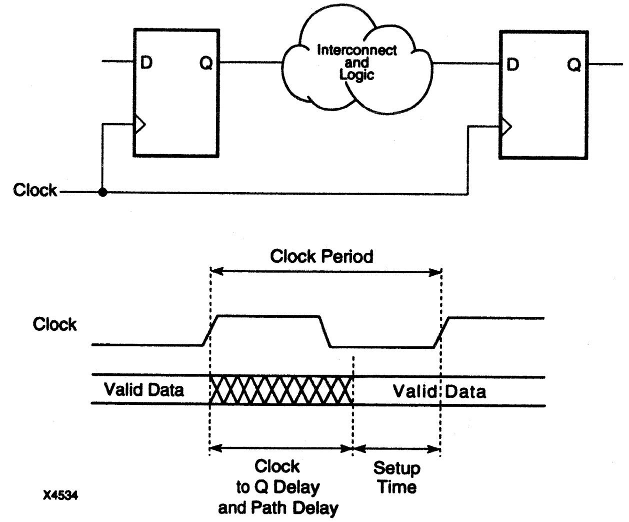


Figure 3-5 Clock To Setup Path, Same Active Clock Edge

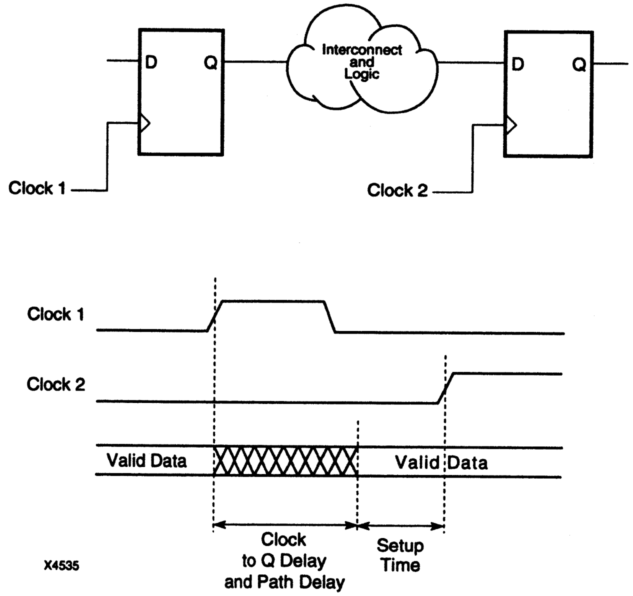


Figure 3-6 Clock To Setup Path, Different Clock Net

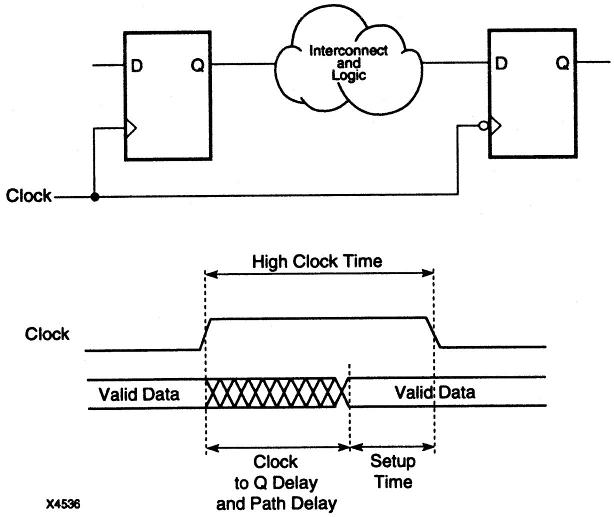


Figure 3-7 Clock To Setup Path, Rising Edge To Falling Edge

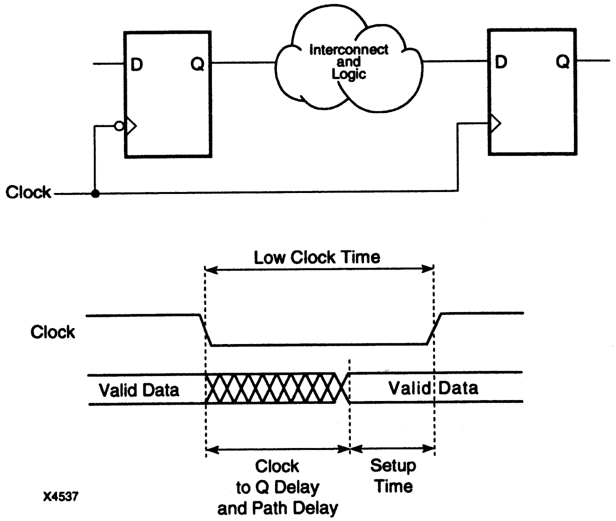


Figure 3-8 Clock To Setup Path, Falling Edge To Rising Edge

–ClockInput

–ClockInput allows you to restrict the report to option paths that end at flip-flop clock pins. These paths can start anywhere. See Figure 3-9.

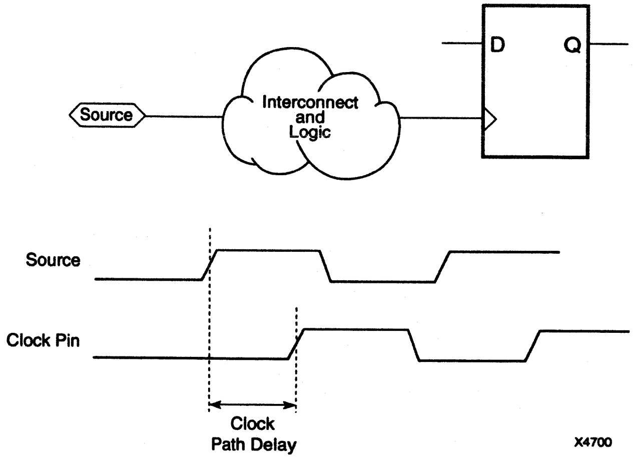


Figure 3-9 Clock Input Path

–Delayless

–Delayless allows you to set a maximum path-delay in nanoseconds. Only paths with delay less than or equal to this value are reported.

–Delaygreater

–Delaygreater allows you to set a minimum path-delay in nanoseconds. Only paths with delay greater than or equal to this value are reported.

–TSMaxpaths

–TSMaxpaths allows you to limit the number of paths reported for each TimeSpec in the XDelay report to a number specified in *path count.*

–Maxpaths

–Maxpaths allows you to limit the number of paths reported. If you use this option while using TimeSpecs, you affect the total number of paths in the system, not the total number for each TimeSpec. This option is also useful when the system memory is limited, since the *path count* also limits how many paths are stored in memory while tracing paths. If XDelay runs out of memory, assigning a Maxpaths *path count* value is the easiest fix.

–Widereport

–Widereport allows you to set the output format to 132 characters per line. This can prevent long block and net names from being truncated in a report.

–Shortreport

–Shortreport allows you to print one delay path per line. It only lists path sources and path loads, without further detailed path information.

–Sort

–Sort allows you to change the order that paths are reported. The default is to order paths from longest to shortest delay. The valid Sort options are listed below.

- **Delaylong** — lists path delays in order from largest to smallest delay (default value).
- **Delayshort** — lists path delays in order from smallest to largest delay.
- **Destblock** — sorts path delays by the name of the destination block. This is useful when trying to group a set of path-delays that all end at one block. It makes it easy to find the worst-case path to each block.
- **Srcblock** — sorts path delays by the name of the source block. This is useful when trying to determine the worst-case path from a given block.
- **Srcclocknet** — sorts path delays by the name of the clock that sources the first element in the path. This is useful when you want to separate delay information for two or more clock nets in one design.
- **Destclocknet** — path delays are sorted by the name of the clock that sources the last element in a path.

–WorstCase

–WorstCase allows you to restricts the display to the path with the longest delay if there is more than one path between a path source and a path destination.

–ClearOptions

–ClearOptions allows you to restore all XDelay options and restores them to their default values.

Analyze

Analyze allows you to get a best-case timing analysis of a design. XDelay computes a maximum toggle rate for each clock net in the system and reports some speed-limiting path information.

Since all on-chip hold times on Xilinx parts are 0, these hold times do not figure into the analysis.

The XDelay timing analysis presents an optimistic view of system timing. There is no adjustment for off-chip delays, so paths that connect to off-chip are probably overly optimistic.

The following requirements apply:

- The clock period for PadToSetup paths and ClockToPad paths must be longer than the maximum path delay, plus any external input timing margin.
- The clock period for ClockToSetup paths (with the same active clock edge) must be longer than the maximum path-delay plus any clock skew between the flip-flops in the path.
- The clock period for ClockToSetup paths (with different clock nets) must be longer than the maximum path delay.
- The clock period for ClockToSetup paths (rising edge to falling edge) must be longer than the maximum path delays plus any clock skew.
- The low time of the clock signal for ClockToSetup paths (falling edge to rising edge) must be longer than the maximum path delay, plus any clock skew.

NOTE *The report produced with the Analyze option describes chip-level performance, not system-level performance.*

Speed

The Speed command allows you to change the current speed grade of your design. You must re-run XDelay to see how changing the speed grade affects design performance.

TimeGroup

The TimeGroup command allows you to create logical groups of objects, such as flip-flops, nets, IOBs, pins, and other groups.

These groups can then be used to create TimeSpecs to help analyze system performance. If you entered timing specifications for XACT-Performance, time groups and timing specifications are automatically created for your design. If you are creating new TimeGroups for a design, the valid TimeGroup types are briefly described below.

- **–FF** a group of flip-flops, specified by its output nets.
- **–Net** a group of all flip-flops that are loads of the specified nets. This allows flip-flops to be grouped by Clock and Clock Enable nets, instead of individually specifying each flip-flop.
- **–IOB** a group of IOBs.
- **–RAM** a group of CLB RAM elements whose outputs drive the named nets.
- **–OutFF** a group of output flip–flops, specified by its IOB name.
- **–RAM_WE** a group of RAM elements that are driven by the specified WriteEnable nets.
- **–Pin** a group of pins. When used as a 'from' group in a TimeSpec, only block output pins are used. When used as a 'to' group, only input pins are used.
- **–Group** a group that contains other groups.

If TimeGroups are created in XDelay, they will be written to the LCA file when it is saved.

DeleteGroup

DeleteGroup allows you to delete specified groups.

QueryGroup

QueryGroup allows you to get a list of the elements of each specified delay group. If you do not specify any groups, it lists all defined groups.

TimeSpec

TimeSpec allows you to set a timing specification in nanoseconds for the paths between the source and load groups.

If the timing specification is not satisfied during design analysis, XDelay issues a warning that the timing specification was not met. When used with the XDelay –FailedPaths option, only those paths that did not meet the required timing specification are in the XDelay report.

If TimeGroups are created in XDelay, they will be written to the LCA file when it is saved.

QuerySpec

QuerySpec allows you to list currently defined TimeSpec. If no TimeSpec is specified, QuerySpec reports all defined TimeSpecs.

DeleteSpec

DeleteSpec allows you to delete TimeSpecs.

QueryTemplate

QueryTemplate allows you to print a list of the current XDelay settings. This is similar to the Settings command in XDM and XDE.

ClearTemplate

ClearTemplate allows you to delete any previously defined XDelay options and restores them to their default.

SaveTemplate

The SaveTemplate command allows you to save the currently defined XDelay options, margins, and Flagblk settings to the named template file. The file extension must be .xtm.

If an XTM file is read that has been created by means other than the SaveTemplate command, the SaveTemplate command might write a new template file that appears different than the original. The new template file is logically equivalent, but the sequence and content of the template file commands can be different. This happens because the contents of the template file are read and translated into an internal non-text form, and are re-translated from this internal form back into text the command.

ReadTemplate

The ReadTemplate command allows you to get a list of XDelay settings from a template file (XTM).

These settings are added to the settings already in memory. If you want to use the file as the only settings, then you first run ClearTemplate.

SetMargins

The SetMargins command allows you to add extra delay onto pad paths that come from, or go to, off-chip. See Figure 3-2 and Figure 3-3.

Adding this extra delay allows you to get a more realistic system clock speed, sine there is always some delay in the path, external to the Xilinx device. SetMargins allows you to add this sort of delay to those paths that travel off-chip.

SetMargins prompts for a path type, which can be either PadToSetup or ClockToPad. It then asks for a clock net to be used as the reference clock where these delays begin or end. Finally, it prompts for the delay margin to add to the internal path delay.

QueryMargins

The QueryMargins command allows you to get all the currently defined Margins.

ClearMargins

The ClearMargins command allows you to remove all defined margins.

Savemargins

The Savemargins command appears on the menu, but is not functional. Use SaveTemplate to save timing margins.

Readmargins

The Readmargins command allows you to read the timing margins that SaveTemplate saved back into the LCA.

Flagblk

The Flagblk command allows you to mark certain blocks to enable or disable some special block delays, when Blkmenus is enabled.

These flags can help decrease the number of false paths in many cases. The following list describes the flags available with Flagblk.

IOB_Enable_O_I

IOB_Enable_O_I allows paths from the IOB O pin to the pad and back into the chip through the IOB I pin.

IOB_Disable_O_I

IOB_Disable_O_I disallows paths from the IOB O pin to the pad and back into the chip through the IOB I pin. This flag is the default.

NOTE *IOB_Enable_O_I and IOB_Disable_O_I only affect bidirectional IOBs with 3-state outputs.*

IOB_Enable_T_I

IOB_Enable_T_I allows paths from the IOB T pin to the pad and back into the chip through the IOB I pin.

IOB_Disable_T_I

IOB_Disable_T_I disallows paths from the IOB T pin to the pad and back into the chip through the IOB I pin. This flag is the default.

TBUF_Enable_I_O

TBUF_Enable_I_O allows paths that go through the TBUF I pin to the TBUF O pin. This flag is the default.

TBUF_Disable_I_O

TBUF_Disable_I_O disallows paths from a TBUF I pin to the TBUF O pin.

TBUF_Enable_T_O

TBUF_Enable_T_O allows paths that go through a TBUF T pin to the TBUF O pin. This flag is the default.

TBUF_Disable_T_O

TBUF_Disable_T_O disallows paths that go through a TBUF T pin to the TBUF O pin.

CLB_Enable_WE

CLB_Enable_WE allows paths through the write-enable input of CLB RAM.

CLB_Disable_WE

CLB_Disable_WE disallows paths through the write-enable input of CLB RAM. This flag is the default.

CLB_Enable_DIN

CLB_Enable_DIN allows paths through DATA inputs of a CLB RAM.

CLB_Disable_DIN

CLB_Disable_DIN disallows paths through DATA inputs of a CLB RAM. This flag is the default.

CLB_Enable_SR_Q

CLB_Enable_SR_Q allows paths through CLB flip-flop Asynchronous Set or Reset inputs. This flag is the default for Asynchronous Set/Reset paths.

CLB_Disable_SR_Q

CLB_Enable_SR_Q disallows paths through CLB flip-flop Asynchronous Set or Reset inputs.

Synchronous

Synchronous allows you to mark the output from all function generators in the selected blocks as synchronous outputs. XDelay treats these outputs like nets marked in XDelay –Breakloop.

This option, however, can flag logic as being synchronous when it is not because the outputs are not treated individually.

Not_Synchronous

Not_Synchronous allows you to clear selected blocks as synchronous outputs.

QueryBlk

The QueryBlk command allows you to get information blocks in the current design. See The XACT Design Editor chapter in this reference guide for more information.

QueryNet

The QueryNet command allows you to get information about nets in the current design. See The XACT Design Editor chapter in this reference guide for more information.

DRC

The DRC command allows you to start the Design Rules Checker (DRC) for the current design. See The XACT Design Editor chapter in this reference guide for more information.

The Misc Menu

This menu contains commands that allow you to use the online help facility, quit out of XDelay, select printer format, escape to the DOS shell (PC only) without quitting XDelay, create a report file, and run a batch file.

Help

The Help command allows you to bring up online help text on specific commands within the XDelay program.

Alternately, you can use the mouse to point at any command or option from the menu, and press the F1 key to bring up the associated help text. Pressing F1 again clears the help text.

Exit

The Exit command allows you to quit from the XDelay program.

Print

The Print command allows you to print all or part of the XDelay display in a format you determine with the Printer command. Two options are available:

- Display prints the display, less the status line and any menus that might be present
- Screen prints the entire display including menus

Printer

The Printer command allows you to determine the format of file that Print creates.

You can change the initial value for the printer type in the xdelay.pro file; the normal initial value is IBMGRAPH. See Table 3-1 for a list of valid printer options.

DOS (For PC only)

The DOS command allows you to temporarily suspend XDelay and open an MS-DOS shell. Type Exit at the shell prompt to return to the XDelay program.

Report

The Report command allows you to redirect the screen output of the XDelay, QueryNet, DRC, or QueryBlk commands to a disk file.

Execute

The Execute command allows you to perform a set of commands from a batch file. Use this command to reduce the number of commands entered redundantly, or to execute a special sequence of commands that are to be performed multiple times.

Table 3-1 List of Printers and Print Formats

Selection	Printer
FX80	Epson FX80
FX85	Epson MX85
FX86	Epson MX86
FX100	Epson FX100
FX185	Epson FX185
FX286	Epson FX286
HPL300	Hewlett Packard Laserjet Printer 300 dpi
HPLASER	Hewlett Packard Laserjet Printer
HPLASER–A4	Hewlett Packard Laserjet Printer–A4
HPLPLUS	Hewlett Packard Laserjet Plus Printer
HPLPLUS–A4	Hewlett Packard Laserjet Plus Printer–A4
HPLPLUSLEGAL	Hewlett Packard Laserjet Plus
HPLPLUSLEGAL–A4	Hewlett Packard Laserjet Plus–A4
IBMgraph	IBM Graphics Printer
MX80	Epson MX80
MX100	Epson MX100
NEC	Nippon Electric Corp.
NECBIG	Nippon Electric Corp. (expanded)
OKI92	Okidata Microline 92
OKI93	Okidata Microline 93
OKI292	Okidata Microline 292
OKI293	Okidata Microline 293
POSTSCRIPT	Postscript Format Printer
RX80	Epson RX80

The Profile Menu

The Profile menu contains commands that allow you to display your XDelay options, save and read your xdelay.pro file, change the menu colors on your color monitor, change your cursor shape, define your mouse buttons, and define your keyboard function keys.

Settings

The Settings command allows you to display the currently defined options within the XDelay program, such as palette default, cursor type, printer type, mouse button configuration, keyboard definitions, etc.

Saveprofile

The Saveprofile command allows you to save the current settings palette color, printer type, keyboard definitions within the program to the xdelay.pro file.

Readprofile

The Readprofile command allows you to read the current settings palette color, cursor type, mouse button configuration into memory from the xdelay.pro file.

Palette

The Palette command allows you to choose a palette from which you select the colors of the menu.

Cursor

The Cursor command allows you to control the shape of the cursor. You can choose an arrow, bug, cross, or gunsight.

Mouse

The Mouse command allows you to define the mouse buttons as Select, Done, Menu or Switch.

Keydef

The Keydef command allows you to define the keyboard function keys as XDelay commands.

Keycursor

The Keycursor command allows the arrow keys to move the cursor through pull-down menus. It also executes the selected option when you press ↵. If Keycursor is off, you must enter commands using the mouse, or by typing the commands via the keyboard.

Text Interface

You can start XDelay from the operating system prompt and use the text interface. The text interface allows you to specify XDelay options in command line syntax.

Using the text interface, you can run template files in batch mode to get delay information about your design.

Syntax

The following command line syntax starts the program.

```
>xdelay [options] design
```

If you do not specify options when you start xdelay at the operating system prompt, you start the graphical interface, not the text interface.

Options

The options available with the text interface are described below.

– help

–help allows you to display the command line help screen.

– d

–d allows you to exit out of XDelay without creating a timing report. Use with the –w option to rewrite the LCA file with net delays without producing a delay report.

– o *output file*

–o allows you to redirect the timing report to the specified output file.

– s

–s allows you to generate the timing report in the short report format.

– t *template file*

–t allows you to specify a template file that XDelay reads for path tracing options. You can create a template file in the graphical user interface, or use an editor to create a template file. See the SaveTemplate and ReadTemplate command descriptions for more information. The template file name must have a .xtm extension.

– timespec

–timespec directs XDelay to compare the implementation of the design with the original XACT–Performance TimeSpec constraints that were defined by the user. The report generated by this option summarizes the paths for each selected TimeSpec and indicated whether the paths passed or failed the timing specification limit.

– u *speed*

–u allows you to set the speed grade for the design.

– w

–w allows you to rewrite the LCA design file after XDelay has retimed all nets in the design.

– x

–x allows you to generate the timing report in the long report format.

Features and Capabilities

This section describes some important features of XDelay.

TimeGroups

TimeGroups are groups of design elements, flip-flops, CLB RAM elements, IOBs, pins, or other TimeGroups. TimeGroups are used in conjunction with timing specification limits in support of XACT-Performance analysis.

You use the TimeSpec command in XDelay to define a timing specification between a pair of TimeGroups. XDelay reports whether the design met the timing specification. You define a 'from_group' (starting point of the analysis), a 'to_group' (ending point of the analysis), and an analysis limit in nanoseconds.

You can use the XDelay –FailedSpec option to limit the timing report to just those paths that failed to meet the timing specifications. A simple example of the report format is shown below:

```
TimeSpec 'slow_clock' summary:
  From  TimeGroup  'slow_counter'
    To  TimeGroup  'blinkenlight'

  TimeSpec limit is : 50.ns   (Spec speed = 20.0 MHZ)
Worst path delay is : 26.2ns  (Real speed = 38.1 MHZ)
  TimSpec passes by : 23.8ns

 List of delay paths:
There are no paths that fail the TimeSpec.
```

Breakloop Option for Combinatorial Logic

You can use the Breakloop option for combinatorial logic, which can help XDelay generate meaningful delay information. This option forces XDelay to treat non-synchronous nets as synchronous. Breakloops mark combinatorial latches as a registered element in the design.

XDelay considers only one feedback loop through the function generator in the time delay estimate. You remove the Breakloop using the –NoBreakloop option.

Figure 3-10 shows a simple circuit and how applying a Breakloop forces XDelay to trace the path. XDelay traces the path starting from the source of the Breakloop Net. The path stops when the Breakloop net is reached. With this option, paths do not travel through the Breakloop source as would normally happen for a combinatorial logic function.

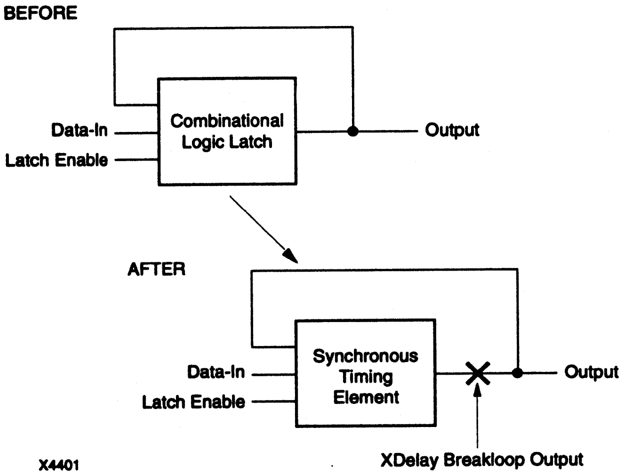


Figure 3-10 –Breakloop Example

Path Delay Searches

You can designate start and end points to narrowly focus the path delay search. XDelay has options that limit where XDelay searches on a specified path.

The –From option allows you to specify the starting limit of the path delay search starting at the named pins, blocks, and/or nets. The wildcard "*" starts searches from all outputs.

The –To option allows you to specify the ending limit of the path delay search ending at the named pins, blocks, and/or nets. The "*" ends searches at all inputs.

You can use the SourceClock and the DestClock options to limit the origins and destinations of delay path searches. These options work similar to the –From and –To options, except that XDelay selects objects by clock net instead of by pin, block, or net.

A clock net is any net that is used as a clock source for flip-flops or latches in a design. Most designs have only one clock net, usually from the global clock buffer. There are occasions where multiple clock nets are used, and you want a different set of data from the delay report about the paths clocked by the different clock nets. The ClockNet features provide some control over this type of search, and should make extracting useful delay information from XDelay easier.

Timing Margins

The Margins commands allow you to specify external delays for paths entering or exiting the chip (PadToSetup and ClockToPad paths). Use these commands to provide XDelay with information about the external system. For example, if a signal enters the chip through an input, it might take 30 ns externally to get to that input. You could set the Margin to 30 ns on that pad so that XDelay includes the external margin in the path delay.

Briefly the Margins commands are:

- SetMargins sets additional external pad timing margins to PadToSetup or ClockToPad paths
- QueryMargins shows a list of all currently defined timing margins
- ClearMargins clears all currently defined margins
- Readmargins reads defined margins from an external file

For more details about these commands, refer to the Graphical Interface section.

Environment Variables

If you are an experienced user of the XDelay program, you can customize your environment to control the XDelay output.

The environment variable XDELAY_PRECISION controls how many digits of precision the XDelay output has. If this environment variable is not used, XDelay uses a single digit of precision.

NOTE

If you change the value of XDELAY_PRECISION, Xilinx does not guarantee accuracy. This variable should only be used to verify floating-point rounding issues.

To set this variable on Unix workstations, type:

```
setenv XDELAY_PRECISION n
```

where *n* is an integer

To set this variable on PCs, type:

```
SET XDELAY_PRECISION n
```

where *n* is an integer.

The environment variable XDELAY_MAX_PATHS limits the number of paths XDelay traces before generating a report. If you reach this limit, you should try focusing the path search to keep XDelay from running out of memory-storing paths.

To set this variable on Unix workstations, type:

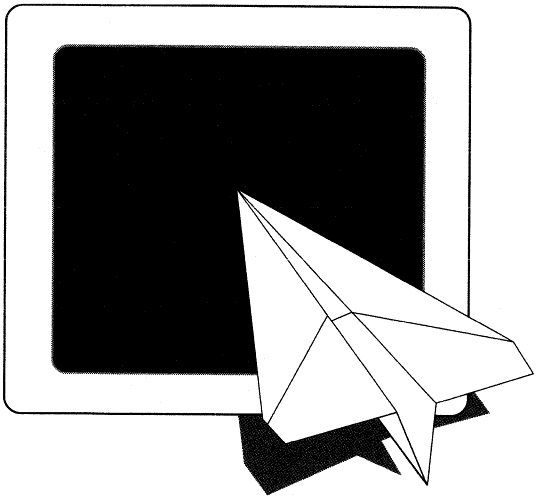
```
setenv XDELAY_MAX_PATHS n
```

where *n* is the maximum number of paths XDelay should trace.

To set this variable on PCs, type:

```
SET XDELAY_MAX_PATHS n
```

where *n* is the maximum number of paths XDelay should trace.



The LCA2XNF Program

XACT Reference Guide, Volume 3

XACT Reference Guide, Volume 3 — April, 1994 (0401131 02)

Printed in U.S.A.

The LCA2XNF Program

This Program is Compatible with the Families Indicated.

☑ XC2000
☑ XC2000L
☑ XC3000
☑ XC3100
☑ XC3000A
☑ XC3000L
☑ XC3100A
☑ XC4000
☑ XC4000A
☑ XC4000H
□ XC7200
□ XC7300

The LCA2XNF program converts an LCA file to an XNF file, which you can use for functional or timing simulation. If the LCA input file is placed and routed, the resulting XNF file contains all worst-case block and net delays for use in timing simulation. You can use the XNF file created by the LCA2XNF program to generate a simulation file for one or more of the Xilinx-supported third-party simulation products. Youis available can also use the output file as a guide for logic partitioning with ADI.

Syntax

The following syntax creates an XNF file from your LCA file.

```
lca2xnf [options] input[.lca] [output[.xnf]]
```

WARNING!!!

You should always specify a different output file name so that the original (unplaced, unrouted) XNF file is not overwritten.

Files

LCA2XNF requires an LCA file as input and generates an XNF file and an SPC file (optionally).

Input Files

input.lca

This is the LCA design input file created by APR, PPR, or the XACT Design Editor (XDE). The LCA input file can be completely routed, partially routed, or unrouted. The LCA extension is optional.

Output Files

output.xnf

The output file is the timing-annotated XNF file. The XNF extension is assumed. If you do not specify an output file name, LCA2XNF uses the input design name and appends the XNF file extension. An overwrite warning displays on the screen if the specified or default output file already exists and was not generated by LCA2XNF.

output.spc

This text file lists the LCA timing specifications for the named device type and speed grade. These tables are similar to those found in *The Programmable Gate Array Data Book* but might represent more current information. Use the –s option to have LCA2XNF create the SPC file.

Options

LCA2XNF has the following options. You can display additional information on each option using the online help feature in XDM.

–b Block Only — Suppress Gate-Modeling Information

The –b option generates an XNF file that contains only block information. A block-only XNF file describes the configuration of each block, such as a CLB or an IOB, but does not include the corresponding gate models for each. It does include the net-delay information for the block pins.

–c Do Not Collapse Split Nets (XC3000A, XC3000L, XC4000 only)

PPR can split a logical net into several physical nets to achieve better routing results. By default, LCA2XNF collapses any split nets back into their original logical net for back-annotation purposes. When you specify this option, LCA2XNF does not collapse the split nets into their original net. In this case, net names might have suffixes, such as, _1 or _2, appended to the name of the original net name.

– e Generate Function Generators and Carry Logic as EQN Symbols

The –e option generates function generators as EQNs. LCA2XNF translates each function generator into one EQN. In addition, if you specify the –v option, LCA2XNF translates each carry logic equation into one EQN. Use the –e option only if the program for which the XNF output file was created can interpret EQNs.

– f Force Timing Models

The –f option forces LCA2XNF to include delay information, even if the LCA design file is not completely routed. The resulting XNF file contains delay models for all blocks and path delays for fully routed nets. For unrouted nets, net delays are 0. With this option, timing simulation can be performed on an LCA file that is not completely routed. If the LCA input file is not completely routed and you do not use the –f option, LCA2XNF issues a warning and generates a unit-delay XNF file. Run XDelay with the –w and –d options to back-annotate net delay information for XC3000A, XC3000L, and XC4000 designs.

– g Generate Gate-Only XNF File

The –g option generates an XNF file that contains only gate models and delay information. A gate-only XNF file does not contain any partitioning information and is similar to an XNF file produced from a schematic. Using an existing LCA design file as input, the –g option creates an XNF file that can be merged with other gate-level XNF files. The –g option is the default.

–m Generate Both Gates and Blocks

The –m option generates an XNF file that contains both gates (DFF, TBUF, GCLK, BUFGP, AND, OR, and so forth), modeling information, and CLBs and IOBs.

–q Generate XC4000 RPM, Compatible with Previous Libraries

The –q option generates an XNF file that contains information to preserve the relative placement of the LCA design. This file is not compatible with the Unified Libraries. Use this option if your design is composed of elements from the previous libraries.

–r Generate XC4000 RPM, Compatible with Unified Libraries

The –r option generates an XNF file that contains information to preserve the relative placement of the LCA design. This file is compatible with the Unified Libraries. Use this option if your design is composed of elements from the Unified Libraries.

– s Print Timing Data Sheets for the Current Design

The –s option generates a text file (output.spc) that lists the LCA timing specifications for the device type and speed grade used in the input LCA design file. LCA2XNF uses the delay information in these sheets, which are similar to those in *The Programmable Logic Data Book*,. However, the timing data sheets can represent more current timing information.

– t Trim Unused CLBs and IOBs

The –t option trims all the unused CLBs and IOBs that were most likely tied as a result of running MakeBits with the tie option. Using this option does not affect simulation. The output XNF file still reflects post-tiedown routing delays. Using this option can generate Warnings 71, 72; or, Warnings 76 and 77 to inform you of an existing tie or unused IOBs and CLBs in the input LCA design file.

NOTE *If you used the MakeBits program with the -t option to save tied nets in your LCA file, you must use the -t option with the LCA2XNF program to remove the tied nets from the XNF file.*

– u Generate Unit-Delay Model

The –u option generates an XNF file without delay information for use in unit-delay simulation. If the input LCA design file is unrouted or partially routed and you did not use the –f option to force timing models, the –u option is the default.

–v Output File in Version 4 XNF Format

The –v option generates a Version 4 XNF file instead of the default Version 5 XNF format. In Version 4, carry logic is represented by combinatorial gates. In Version 5, carry logic is represented by carry symbols. You should not use this output file as input to PPR.

– w Suppress Overwrite Warning

The –w option prevents LCA2XNF from issuing a warning message before writing over an existing XNF file not generated by LCA2XNF. By default, LCA2XNF issues a warning and requests permission to overwrite.

Warning Messages and Recovery Techniques

This section describes the warning messages that LCA2XNF can generate. An explanation and workaround solution follows each warning message.

Warning 11 **Input ignored.**

This indicates a corrupt LCA design file. Verify that there is adequate disk space when the LCA design file is created, and that the program that created it, PPR, APR, XDE, and so forth, does not report any errors.

Warning 23 **Design is unrouted. Unit delay model will be generated.**

If any nets in the LCA are unrouted and the –f option is not used, LCA2XNF issues this warning. In this case, LCA2XNF generates a unit-delay model since all delays cannot be modeled. If the LCA file is from PPR, use XDelay with the –d and –w options to add delay information to the LCA file. Use XDE to check that the design is fully routed, or use the –f option in LCA2XNF to force a timing model.

Warning 24 **No speed grade specified. Unit delay model will be generated.**

If there is no speed grade specified in the input LCA design file. In this case, LCA2XNF generates a unit-delay model since no delays can be modeled. Use XDE to check that a speed grade is specified.

Warning 25 **Invalid speed grade specified.**

The speed grade specified is not valid for the given part type. The legal speed grades for each part type are shown in *The Programmable Logic Data Book*, or you can find the speed grades by using the Speed command in XDE.

Warning 26 **Pin** *pin_name* **used but no net connected to it.**

A pin is configured inside a CLB or IOB, but no net is connected to that pin. LCA2XNF creates a net on that pin with the same name as the pin itself, that is, *block_name.pin_name,* which results in a loadless or sourceless net in the XNF file.

Warning 27 **Unknown command line flag** *flag* **ignored.**

An illegal option was specified on the command line. Check the LCA2XNF usage and options section at the beginning of the chapter for more information on legal option flags.

Warning 28 **Extra argument** *argument* **ignored.**

A third parameter was found on the command line after the output XNF file name. This extra parameter is ignored.

Warning 50 **Signal** *signal_name* **is loadless.**

The indicated signal is loadless. Check the LCA design file and the output files of the programs that created the LCA design file to determine why the signal is loadless.

Warning 51 **No configuration found in block** *block_name***.**

The indicated block (CLB, IOB, and so forth) has no Config statement. Check and fix the indicated CLB.

Warning 53 **No Pull-ups found on the Decode long line net** *netname***.**

All decode longlines must be connected to pull-ups. Check the design and add the pull-ups.

Warning 54 **Signal** *signal name* **is sourceless.**

Check the LCA design file and the output files of the programs that created the LCA design file to determine why the signal is sourceless.

Warning 61 **DX is used but QX is not in block** *block_name***.**

A CLB flip-flop data pin is used, but the output is not.

Warning 62 **DY is used but QY is not in block** *block_name***.**

A CLB flip-flop data pin is used, but the output is not.

Warning 63 **FFX:K is used but QX is not in block** *block_name***.**

A CLB flip-flop clock input is used but the output is not. Check the CLB configuration in XDE.

Warning 64 **FFY:K is used but QY is not in block** *block_name***.**

A CLB flip-flop clock input is used but the output is not. Check the CLB configuration in the XACT Design Editor.

Warning 71 **CLB** *block_name* **is removed since all output pins are loadless.**

The indicated CLB has no loads. Check the LCA design file and the output files of the programs that created the LCA design file to determine why the CLB is loadless.

Warning 72 **IOB** *block_name* **is removed since it is unconnected or unconfigured.**

The indicated IOB has no connections or has no configuration. Check the LCA design file and the output files of the programs that created the LCA design file to determine why the IOB is loadless.

Warning 73 **Unused PULLUP** *pullup_name* **is removed.**

The indicated pull-up was not connected to a horizontal Longline or an IOB, and was removed. Check the LCA design file and the output files of the programs that created the LCA design file to determine why the pull-up is loadless.

Warning 74 **GCLK/ACLK shared the pad with IOB** *block_name* **at LOC=***LOC_number***.**

The global and alternate clock buffers (GCLK/ACLK) for XC2000/XC3000 devices can be sourced from IOBs or internal logic. This warning occurs if the IOB sourcing GCLK or ACLK has other signals attached to it and lets you know that there could be signals interfering with your clock signal. This error is most commonly seen on the XC3000 family when using direct clock inputs (TCLKIN and BCLKIN). Other IOBs could have been inadvertently placed on the special IOBs that contain TCLKIN and BCLKIN.

Warning 76 **CLB** *block_name* **does not have any loaded pins.**

The indicated CLB has no loads. Check the LCA design file and the output files of the programs that created the LCA design file to determine why the CLB is loadless.

Warning 77 **IOB** *block_name* **is not connected or configured.**

The indicated IOB has no connections or has no configuration. Check the LCA design file and the output files of the programs that created the LCA design file to determine why the IOB is loadless.

Warning 78 **Nodes** *node* **and** *node* **are in loop via net** *net***.**

The nets at the indicated nodes are actually the same net and form a loop. Check the LCA design file to determine why the loop is there and remove it.

Error Messages and Recovery Techniques

This section describes the error messages that LCA2XNF can generate. An explanation and workaround solution follows each error message.

Error 1 **Unknown block** *block_name* **referenced.**

An invalid block name was found in the LCA design file. This should happen only when the input LCA design file was created after the MAP2LCA program could not fit a design into the target LCA part type. You must either reduce the design or use a larger part type.

Error 2 **Multiple connections to pin** *pin_name* **by nets** *net_name net_name* **....**

This indicates a corrupted LCA design file. Verify that there is adequate disk space when the LCA design file is created, and that the program that creates it, APR/PPR, XDE, and so forth, does not report any errors.

Error 3 **Invalid pin** *pin_name* **for block** *block_name*.

This indicates a corrupted LCA design file. Verify that there is adequate disk space when the LCA design file is created, and that the program that creates it, APR/PPR, XDE, and so forth, does not report any errors.

Error 4 **Invalid type** *type_name* **on Base record for block** *block_name*.

There is an illegal value on a Base record for the specified CLB or IOB. This error typically results from an incorrectly specified CLB or IOB primitive on a schematic. Correct the Base record on the schematic and generate the LCA design file again.

Error 5 **Multiple Base/Config records for block** *block_name*.

Only one Base record and one Config record are allowed on a single block. This error is typically caused by an incorrectly specified CLB or IOB primitive on a schematic. Edit the schematic to use only one Config and one Base record, and generate the LCA design file again.

Error 6 **Config/Equate record not allowed for block** *block_name*.

A Config or Equate record was found for a block that is not configurable. This indicates a corrupted LCA design file. Verify that there is adequate disk space when the LCA design file is created, and that the program that creates it, PPR/APR, XDE, etc., does not report any errors.

Error 7 **Invalid tag** *tag_name* **or setting** *setting_name* **on CONFIG/Equate record** *block_name*.

An invalid Config record was found. This is typically caused by an incorrectly specified CLB or IOB primitive on a schematic. Correct the Config or Equate record on the schematic, and generate the LCA design file again.

Error 8 **No Latch/FF option selected for CLB** *CLB_name*.

The Config record for the CLB does not specify if the Latch or FF option should be used for a XC2000 design. Since there is no default value, it must be explicitly specified. Check the CLB primitive on the schematic for either Q:Latch or Q:FF in the Config record. Generate the LCA design file again.

Error 9 **Invalid equation in block** *block_name*.

The indicated equation is not valid. This typically results from an incorrectly specified CLB symbol on a schematic. Edit the schematic to correct the Equate parameter, and generate the LCA design file again.

Error 12 **Invalid pin name** *pin_name*.

An invalid pin name was found. This error indicates a corrupted LCA design file. Verify that there is adequate disk space when the LCA design file is created, and that the program that creates it, PPR/APR, XDE, and so forth, does not report any errors.

Error 13 **Missing part type on Design record.**

The Design record in the LCA design file does not specify the part type of the design. This indicates a corrupted LCA design file. Verify that there is adequate disk space when the LCA design file is created, and that the program that creates it, APR/PPR, XDE, and so forth, does not report any errors.

Error 14 **Base/Config/Equate record with no current block defined.**

A configuration record was found outside of an EditBlk-EndBlk record group. This indicates a corrupted LCA design file. Verify that there is adequate disk space when the LCA design file is created, and that the program that created it, APR/PPR, XDE, and so forth, does not report any errors.

Error 15 **Editblk/Endblk records don't match.**

This indicates a corrupted LCA design file. Verify that there is adequate disk space when the LCA design file is created, and that the program that created it, APR/PPR, XDE, and so forth, does not report any errors.

Error 16 **Part type not specified in LCA design file.**

The part type is not specified in the design record in the LCA design file. This indicates a corrupted LCA design file. Verify that there is adequate disk space when the LCA design file is created, and that the program that created it, APR/PPR, XDE, and so forth, does not report any errors.

Error 17 **Unable to open data file** ***filename*****.**

The system file containing data about parts, packages, and delays could not be opened. Repeat the installation procedure to install the LCA2XNF data files properly.

Error 18 **Part type** ***part_type*** **not found in data file** ***filename*****.**

The part type specified in the LCA design file was not found in the system data file. This error might indicate a corrupted LCA design file. Verify that there is adequate disk space when the LCA design file is created, and that the program that created it, APR/PPR, XDE, and so forth, does not report any errors. If the problem persists, repeat the installation procedure to install the LCA2XNF data files properly. Also, the specified part type could be a new part/package. In this case, install the new part/package update.

Error 19 **Block/Die/Alias not found in data file** ***filename*****.**

The system file containing data about parts, packages, and delays might be corrupted. Repeat the installation procedure to install the LCA2XNF data files properly. Also, the specified part type could be a new part/package. In this case, install the new part/package update.

Error 20 **Unable to open output file** ***filename*** **for writing.**

Verify that there is adequate disk space, that the target disk is not write-protected, and that a read-only file with the target name does not exist.

Error 21 **Out of memory.**

Memory requirements are dependent on the device used and the number of memory resident programs running. Table 3-2 lists the suggested memory of each family.

Table 3-2 Memory Requirements

Device Family	Memory Requirements
XC2000 Family	2.5 MB
XC3000, XC3100 Family	6.5 MB
XC3000A/L Family	6.5 MB
XC4000 through XC4010	12.0 MB

Error 22 **Unable to open LCA design file** *filename* **for reading.**

The LCA input file could not be found. Check the file name and execute the LCA2XNF program again.

Error 29 **Base F selected and G equation specified on CLB** *CLB_name*.

An illegal CLB configuration is specified. If a CLB is configured as Base F, no equation is allowed for G. This error typically results from an incorrectly specified CLB symbol on a schematic. Edit the schematic to remove the G equation, and generate the LCA design file again.

Error 30 *option 1* **and** *option 2* **cannot be used together.**

Mutually exclusive options are:
-b and -g
-b and -m
-g and -m
-f and -u
-q and -v
-r and -v
-q and -r

Error 31 **Invalid LCA design file.**

The LCA design file does not contain a valid design record. This errir might indicate a corrupted LCA design file. Verify that there is adequate disk space when the LCA design file is created, and that the program that creates it, APR/PPR, XDE, and so forth, does not report any errors.

Error 32 **Error while writing XNF information to disk.**

Verify that there is adequate disk space, that the target disk is not write-protected, and that a read-only file with the target name does not exist.

Error 33 **Unable to rename the temp file to the output XNF file because the disk is full.**

Free some disk space and execute the LCA2XNF program again.

Error 34 **Unable to open the output file for writing timing data sheets when option -s is specified.**

Free some disk space and execute the LCA2XNF program again.

Error 36 **-q/-r options cannot be used on part type** ***parttype*****.**

These two options can only be used with the XC4000 family.

Error 50 **Memory missing Setmemory for block** ***block_name*****.**

An XC4000 CLB function generator was configured as a memory, but the Setmemory statement was omitted. Check if the program that created the LCA design file reported any errors.

Error 51 **Value missing in Setmemory for block** ***block_name*****.**

An XC4000 CLB function generator was configured as a memory, but the Setmemory value was omitted. Check if the program that created the LCA design file reported any errors.

Error 52 **Prefix x missing in Setmemory for block** ***block_name*****.**

An XC4000 CLB function generator was configured as a memory, but the Setmemory X prefix was omitted. Check the program, which created the LCA design file for any errors.

Error 53 **Value has too many digits for Setmemory F/G/FG for block** ***block_name*****.**

An XC4000 CLB function generator was configured as a memory, but the Setmemory value was too long. The Setmemory values can contain four or eight hex characters. Check if the program that created the LCA design file reported any errors.

Error 54 **Mnemonic missing in Configcarry for block** ***block_name*****.**

An XC4000 CLB configured to use the carry logic must have a mnemonic selected that describes the configuration of the carry logic. This mnemonic is missing from the Configcarry statement. Check if the program that created the LCA design file reported any errors.

Error 55 **Illegal mnemonic found in Configcarry for block** *block_name*.

An XC4000 CLB configured to use the carry logic must have a legal mnemonic selected that describes the configuration of the carry logic. The mnemonic found on the Configcarry statement is illegal. The valid values are listed in "The XACT Design Editor" in the *XACT Reference Guide* and in XDE itself under the ConfigCarry command. Check if the program that created the LCA design file reported any errors.

Error 80 **Can't open partlist.xct.**

LCA2XNF cannot find the partlist.xct file. Check for the existence of this file in the $\XACT\data directory. Also check the XACT environment variable.

Error 81 **Can't find parttype** ***parttype*** **in the patlist file.**

Make sure the selected parttype is correct in the LCA file. Check for the existence of this file in the $\XACT\data directory. Also check the XACT environment variable.

Error 82 **Can't open package file** *package_filename*.

LCA2XNF cannot open the indicated package file.

Check for the existence of this file in the $\XACT\data directory. Also check the XACT environment variable.

Error 83 **Illegal revision number** *number* **found in partlist.xct.**

The correct revision of partlist.xct is provided with the latest official release. Check the release/install procedures.

Error 84 **Invalid format detected in** *speeds file name*.

The correct format is provided with the latest official release. Check the release/install procedures.

Error 101 **OQ is selected but not OK in block** *block_name*.

The registered output of an IOB flip-flop is used but the clock is not connected. Check and fix the output flip-flop in the indicated IOB.

Error 102 **Both Out O and OQ are selected in block** *block_name*.

Both the direct and registered outputs of the indicated IOB are enabled. IOBs can have either registered or direct outputs but not both. Check and fix the indicated IOB.

Error 103 **OKNOT is selected but OK is not in block** *block_name*.

The output clock inversion is selected, but the output clock itself is not selected. Both must be selected. Check and fix the indicated IOB.

Error 104 **Both 0 Set and Reset are selected in block** *block_name*.

An XC4000 IOB output register (OUTFF or OUTFFT) can be configured as either Set or Reset, but not both, on power-up or when the global Set/Reset signal is asserted. Check and fix the indicated IOB.

Error 105 **TRI:NOT is selected but TRI:T is not in block** *block_name*.

The output 3-state sense inversion is selected, but the output 3-state buffer is not selected. Both must be selected. Check and fix the indicated IOB.

Error 106 **Both PULLUP and PULLDOWN are selected in block** *block_name*.

XC4000 IOBs have both pull-up and pull-down resistors, but they cannot both be enabled. Enable either the pull-up or the pull-down resistors. Check and fix the indicated IOB.

Error 107 **Both Out INFF Set and INFF Reset are selected in block** *block_name*.

An XC4000 IOB input register (INFF, INLAT, or INREG) can be configured as either Set or Reset, but not both, on power-up or when the global Set/Reset signal is asserted. Check and fix the indicated IOB.

Error 108 **IKNOT is selected but IK is not in block** *block_name*.

The input clock inversion is selected, but the input clock itself is not selected. Both must be selected. Check and fix the indicated IOB.

Error 109 **Both I and IQ are selected for I1 in block** *block_name*.

An XC4000 IOB has two inputs into the LCA, I1 and I2. Each input can each receive one of three signals: the direct input (I), the latched input (IQL) or the registered (IQ) input. In the configurations above, the indicated input signals are both connected to I1. Only one of these can be connected to I2. Check and fix the indicated IOB.

Error 110 **Both IQ and IQL are selected for I1 in block** *block_name*.

An XC4000 IOB has two inputs into the LCA, I1 and I2. Each input can each receive one of three signals: the direct input (I), the latched input (IQL) or the registered (IQ) input. In the configurations above, the indicated input signals are both connected to I1. Only one of these can be connected to I2. Check and fix the indicated IOB.

Error 111 **Both IQL and I are selected for I1 in block** *block_name*.

An XC4000 IOB has two inputs into the LCA, I1 and I2. Each input can each receive one of three signals: the direct input (I), the latched input (IQL) or the registered (IQ) input. In the configurations above, the indicated input signals are both connected to I1. Only one of these can be connected to I2. Check and fix the indicated IOB.

Error 112 **Both I and IQ are selected for I2 in block** *block_name*.

An XC4000 IOB has two inputs into the LCA, I1 and I2. Each input can each receive one of three signals: the direct input (I), the latched input (IQL), or the registered (IQ) input. In the configurations above, the indicated input signals are both connected to I2. Only one of these can be connected to I2. Check and fix the indicated IOB.

Error 113 **Both IQ and IQL are selected for I2 in block** *block_name*.

An XC4000 IOB has two inputs into the LCA, I1 and I2. Each input can each receive one of three signals: the direct input (I), the latched input (IQL), or the registered (IQ) input. In the configurations above, the indicated input signals are both connected to I2. Only one of these can be connected to I2. Check and fix the indicated IOB.

Error 114 **Both IQL and I are selected for I2 in block** *block_name*.

An XC4000 IOB has two inputs into the LCA, I1 and I2. Each input can each receive one of three signals: the direct input (I), the latched input (IQL), or the registered (IQ) input. In the configurations above, the indicated input signals are both connected to I2. Only one of these can be connected to I2. Check and fix the indicated IOB.

Error 115 **Both F and H are connected to X in block** *block_name*.

An XC4000 CLB has three function generators: F, G, and H. Only one output from either the F or H function generator can be connected to the X output. In this configuration, both F and H are connected to the X output. Check and fix the indicated CLB.

Error 116 **Both G and H are connected to Y in block** *block_name*.

An XC4000 CLB has three function generators: F, G, and H. Only one output from either the G or H function generator can be connected to the Y output. In this configuration, both G and H are connected to the Y output. Check and fix the indicated CLB.

Error 117 **Both QX and DIN are connected to XQ in block** *block_name*.

An XC4000 CLB has two connections to the XQ pin: QX from the output of the CLB flip-flop and DIN when the special signal-buffering feed-through path is used. The flip-flop output and the feed-through path cannot be used together. Check and fix the indicated CLB.

Error 118 **Both QY and EC are connected to YQ in block** *block_name*.

XC4000 CLB has two connections to the YQ pin: QY from the output of the CLB flip-flop and EC when the special signal-buffering feed-through path is used. The flip-flop output and the feed-through path cannot be used together. Check and fix the indicated CLB.

Error 119 **Multiple pins are connected to H1 in block** *block_name*.

The indicated pin can be sourced by only one signal. Check and fix the indicated CLB.

Error 120 **Multiple pins are connected to DIN in block** *block_name*.

The indicated pin can be sourced by only one signal. Check and fix the indicated CLB.

Error 121 **Multiple pins are connected to SR in block** *block_name*.

The indicated pin can be sourced by only one signal. Check and fix the indicated CLB.

Error 122 **Multiple pins are connected to EC in block** *block_name*.

The indicated pin can be sourced by only one signal. Check and fix the indicated CLB.

Error 123 **Multiple pins are connected to DX in block** *block_name*.

The indicated pin can be sourced by only one signal. Check and fix the indicated CLB.

Error 124 **Multiple pins are connected to DY in block** *block_name*.

The indicated pin can be sourced by only one signal. Check and fix the indicated CLB.

Error 125 **FFX:K is used but FFX K is not in block** *block_name*.

The indicated CLB flip-flop is enabled, but no clock is enabled. Check and fix the indicated CLB.

Error 126 **FFY:K is used but FFY K is not in block** *block_name*.

The indicated CLB flip-flop is enabled, but no clock is enabled. Check and fix the indicated CLB.

Error 127 **FFX:K is used but DX is not in block** *block_name*.

The indicated CLB flip-flop is enabled, but has no input. Check and fix the indicated CLB.

Error 128 **FFY:K is used but DY is not in block** *block_name*.

The indicated CLB flip-flop is enabled, but has no input. Check and fix the indicated CLB.

Error 131 **Both FFX Set and Reset are selected in block** *block_name*.

The indicated CLB flip-flop has both Set and Reset enabled. XC4000 CLB flip-flops can be either set or reset on power-up or by asserting either the Global Set or Global Reset signal, but not both. Check and fix the indicated CLB.

Error 132 **Both FFY Set and Reset are selected in block** *block_name*.

The indicated CLB flip-flop has both Set and Reset enabled. XC4000 CLB flip-flops can be either set or reset on power-up or by asserting either the Global Set or Global Reset signal, but not both. Check and fix the indicated CLB.

Error 133 **Both F and FG are selected for RAM in block** *block_name*.

When an XC4000 CLB F function generator is configured as a RAM, the CLB base can be F or FG, not both. Check and fix the indicated CLB.

Error 134 **Both G and FG are selected for RAM in block** *block_name*.

When an XC4000 CLB G function generator is configured as a RAM, the CLB base can be G or FG, not both. Check and fix the indicated CLB.

Error 138 **Pin** *pin_name* **not found for the carry mode in block** *block_name*.

Each carry logic mode implies a Boolean equation in terms of CLB input pins. Those pins must be connected.

Error 145 **F is loaded but not sourced in block** *block_name*.

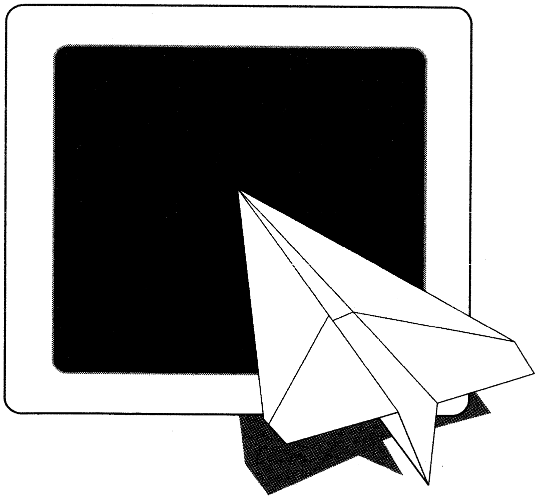
The indicated function generator has an output, but has no inputs. Check and fix the indicated CLB.

Error 146 **G is loaded but is not sourced in block** *block_name*.

The indicated function generator has an output, but has no inputs. Check and fix the indicated CLB.

Error 147 **H is loaded but is not sourced in block** *block_name*.

The indicated function generator has an output, but has no inputs. Check and fix the indicated CLB.



The XNFBA Program

XACT Reference Guide, Volume 3

Printed in U.S.A.

The XNFBA Program

This Program is Compatible with the Families Indicated.

☑ XC2000
☑ XC2000L
☑ XC3000
☑ XC3100
☑ XC3000A
☑ XC3000L
☑ XC3100A
☑ XC4000
☑ XC4000A
☑ XC4000H
□ XC7200
□ XC7300

The XNFBA program combines one of the following the pre-route XG/XFF/XNF files and the post-route XNF file into a new file which has the original symbol and signal names with post-route delays.

For all families, you must keep all design files consistent. If you change anything in the schematic, rerun either APR or PPR before using XNFBA. If you use XDE on the design that was routed by APR or PPR, XNFBA might not work properly.

For all families, if you run the MakeBits program with the –t option to save tied nets in your LCA file, you must use the –t option with the LCA2XNF program to remove the tied nets from the XNF file.

If you want to use CLB and IOB symbols in your schematic, you must place and route your design. Placing and routing your design back-annotates the correct simulation models.

For XC2000 and XC3000 families, ensure that the MAP2LCA program does not generate the AKA file, since the AKA file can cause undesirable back-annotation results.

For XC3000A, XC3000L, XC3100, and XC4000 families, there can be a difference in the timing delays between the XNF file that the LCA2XNF program generates and the XNFBA output file. This difference is due to the logic replication that PPR performs. For efficient partitioning, PPR might replicate a combinatorial gate into a few gates. XNFBA combines the replicated gates back into the original gates. However, the replicated gates might not have identical routing delays on their input pins. XNFBA chooses the longest delay to generate a worst-case timing model.

NOTE

XNFBA does not restore names that were shortened by the MAP2LCA program. Therefore, do not use MAP2LCA –s if you plan to perform timing simulation.

Syntax

The following syntax creates an annotated XNF file from your pre-route and post-route XNF files.

```
XNFBA pre-route XNF [.xg, .xff, .xnf] post-route XNF
[.xnf] [options]
```

Files

XNFBA requires two input files and can generate three output files.

Input Files

XNFBA requires two input files: the pre-route XNF file and the post-route XNF file.

The pre-route XNF file must be flattened prior to using APR or PPR. If you have hierarchical XNF files, you can run XNFMerge to flatten them. XNFMerge produces an XFF output file. The XG file that XBLOX creates is also an acceptable input file. If you do not specify an extension with the input file name, XNFBA looks for a root file name with either an .xg, .xff, or .xnf extension, in that order.

The post-route XNF file is required. Use LCA2XNF with the –g option to generate this file. The XNF extension is optional when specifying the XNF input file. The design should have been completely routed with no DRC errors.

The input file that runs LCA2XNF is a routed LCA file. For XC2000 and XC3000 designs, APR generates the LCA file. For XC3000A, XC3000L, XC3100, and XC4000 designs, PPR generates the LCA file; running XDelay with the –d and –w options adds routing delay information to the LCA file.

Output Files

XNFBA outputs the following three files:

xnfba.xnf

This file contains delay information, and net and instance names from the original schematic, which are used as input for simulation. Use the –o option to specify an output file name if you do not want the default file name, xnfba.xnf.

post-route file root.mba

XNFBA creates an MBA file when you use the –m option. This is a Mentor back-annotation file.

xnfba.rpt

XNFBA creates an RPT file that is a report file. You can specify the report file name when you use the –r option. The default file name is xnfba.rpt.

Options

XNFBA has three options.

–m Write Mentor Back-Annotation File

This option allows you to generate a Mentor back-annotation file.

–o Specify XNF Output File Name

This option allows you to specify the name of the output XNF file. The default is xnfba.xnf.

–r Specify XNF Report File Name

This option allows you to specify the report file name. The default is xnfba.rpt.

Warning Messages

Error and warning messages that are numbered below 200 are the same messages that XNFMAP and XNFMerge issue. The following warnings and errors are specific to XNFBA.

Warning 0-200 **See XNFMAP and XNFMERGE chapters in the** *XACT Reference Guide***.**

Warning 201 **Unknown flag** *flag* **is ignored.**

Check your command line and try again, if necessary.

Warning 202 **Extra argument** *argument* **ignored.**

Check your command line and try again, if necessary.

Warning 203 **The two input XNF files are identical. No operation will be performed.**

The file names specified for the pre-route and post-route XNF are the same. Check your command line and try again.

Warning 205 **The post-route XNF file contains no timing information.**

The output of XNFBA does not contain any timing information. If the design is routed by PPR, make sure you have run XDelay to add routing delays to the LCA file before running LCA2XNF.

Error Messages and Recovery Techniques

Error 251 **The pre-route XNF file is required but is not available.**

This message could indicate either problems with entries on the command line, or the file location or existence. Check your command line and try again.

Error 252 **The post-route XNF file is required but is not available.**

This message could indicate either problems with entries on the command line, or the file location or existence. Check your command line and try again

Error 253 **CLB/IOB detected in** *post-route xnf file*. **Please run LCA2XNF -g and try again.**

XNFBA requires a post-route XNF file that was generated by LCA2XNF with the –g option. The –g option ensures that the CLB and IOB symbols are not present in the XNF file.

Error 254 **Invalid part type** *parttype* **found in file** *file name*.

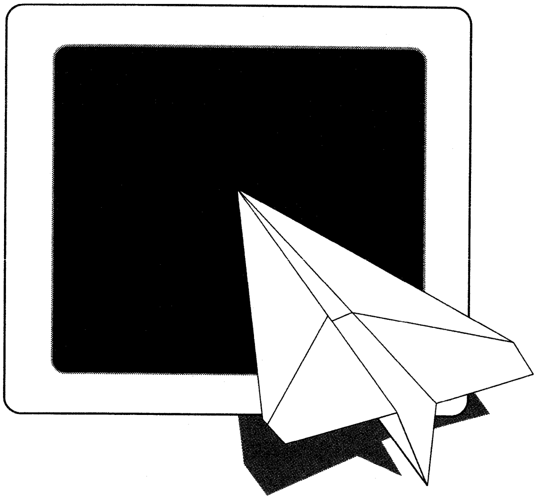
Part types must be valid XC2000, XC3000, XC3000A/L, XC3100, or XC4000 part types. Check your design and try again.

Error 255 **The pre-route and post-route XNF files have inconsistent part types:** *parttype* **and** *parttype*.

The pre-route and post-route XNF files must be from the same family (XC2000, XC3000, XC3000A/L, XC3100, or XC4000). Check your design and try again.

Error 256 **The pre-route XNF file contains symbol** *symbol type symbol name* **which could not be back-annotated.**

If the symbol in question is a hierarchical design, run XNFMerge to flatten the design before routing. If the symbol in question is a CLB or IOB symbol, replace it with the equivalent gate-level symbol, if possible. If the symbol in question is a CLB or IOB symbol and it is not possible to replace it with the equivalent gate-level symbol, you can use the XTF file produced by XNFPrep as the pre-route XNF file. The XTF file contains equivalent gate-level symbols for CLB and IOB symbols. However, the XTF file does not contain the original symbols and signals trimmed away by XNFPrep. In general, avoid using CLB and IOB symbols if you plan to do back-annotation.



XACT Reference Guide, Volume 3

The XACT Design Editor

XACT Reference Guide, Volume 3 — April, 1994 (0401131 02)

Printed in U.S.A.

The XACT Design Editor

This Program is Compatible with the Families Indicated.

☑ XC2000
☑ XC2000L
☑ XC3000
☑ XC3100
☑ XC3000A
☑ XC3000L
☑ XC3100A
☑ XC4000
☑ XC4000A
☑ XC4000H
□ XC7200
□ XC7300

The XACT Design Editor (XDE) contains several programs that allow you to make changes to an existing design, or create a new design. EditLCA is a device-level graphical editor for your LCA designs. DRC is a device-level design rules checker for LCA designs. MakeBits is a program that generates a bitstream from an LCA design that you can download to a device. Finally, MakePROM is a program that reformats bitstream data for programming into PROMs.

The XDE Executive is the top level of the XDE program. The XDE Executive contains commands to perform the following functions:

- Load and save design files
- Start any XDE subprogram
- Change the XDE XDE Executive user-definable options

Syntax

To run the XDE program, type the following command at the operating system prompt to immediately display the XDE Executive.

```
> xde
```

To run the XDE program from within the XACT Design Manager, type the xde command with the desired options on the command line.

```
> xde options
```

XDE Executive Display

The XDE Executive display consists of the following:

- Current settings of XDE Executive options
- Pull-down menus (Programs, Designs, and Profile)
- Cursor
- Message line
- Input line

The XDE Executive display shows the settings of four options that are described below.

- **Directory** — This is the directory to which the XDE program has access.
- **Design** — This is the current design file used by the XDE program to perform any function requiring a design file, such as starting the Design Editor.
- **Part** — This is the part type of the LCA design that specifies the family and package.
- **Mouse** — This is the serial port of the PC to which the mouse is connected, eitther COM1 or COM2. It is not necessary to set other parameters, such as the baud rate.

When you start the XDE program, the initial values for these settings (except directory) are automatically loaded from the xact.pro file. The XDE Executive contains commands to alter these settings, either for one session only or for subsequent sessions by changing the xact.pro file. If more than one xact.pro file exists in the system, the XDE program uses the xact.pro file in the directory from which the XDE program is started.

Options

You have several options for starting the XDE program.

–e *file name*

This option directly enters EDITLCA (not at executive level) for the specified LCA file.

–g *mode*

This option turns on graphics. Table 3-3 lists the valid modes.

Table 3-3 XDE Graphic Modes (PC only)

CGA	Original PC CGA mode
EGA4	EGA, four colors
EGA8	EGA, eight colors
EGA16	EGA, 16 colors
VGA4	640 x 480 VGA, four colors
VGA8	640 x 480 VGA, eight colors
VGA16	640 x 480 VGA, 16 colors
VGAP4	800 x 600 VGA, four colors
VGAP8	800 x 600 VGA, eight colors
VGAP16	800 x 600 VGA, 16 colors

–lcaver=*number*

This option controls the output LCA file version where *number* is a positive integer. For example, if *number* is 1, XDE writes LCA files in the old LCA file format. This format has no PIP direction information, no VERSION record, and no net names on PROGRAM and NPROGRAM records. If *number* is 2 or greater, XDE writes the LCA files with a VERSION number, PIP direction information, and net names on PROGRAM and NPROGRAM records.

–m

This option sets the mouse COM port to 1 or 2 for PCs only.

–n

This option turns off the floating-point processor.

–t

This option turns off graphics.

–p

This option sets the size of the graphics buffer using the screen size.

–pc *x:y*

This option sets the size of the graphics buffer large enough for an *x* by *y* array of CLBs.

–pd

This option sets the size of the graphics buffer using the die size.

–px *n:m*

This option sets the size (*n* by *m*) pixel graphics buffer.

Loading and Saving Design Files

The XDE Executive Read command loads data from the current design file into memory. The Save and File commands can write data from memory into a design file; File also leaves the XDE program and returns to the operating system prompt. The Read, Save, and File commands are in the Designs menu.

With the Fastsave command you can save your designs in a non-portable binary form, which is read much more quickly than an LCA file. This command is in the Profile menu and is for PCs only.

Starting XACT Subprograms from XDE

The Programs menu in the XDE Executive contains commands for starting the Design Editor, using the Design Rules Checker, making bitstreams, and generating PROM files. The subprogram operates on the design file that is currently loaded. If you have not loaded a design, the XDE program loads the current design file, as specified in the XDE Executive, before starting the selected subprogram. XDE prompts for a design when you do not specify one.

Changing XDE Executive Settings

There are several options available for changing the XDE Executive environment:

Beep	Control beep noise.
Cursor	Control the cursor shape.
Design	Set the current design file.
Directory	Set the current directory.
Fastsave	Enable or disable fast-design-file read/write operations on PCs only.

Keycursor Use the arrow keys to move the cursor.

Keydef The definition for the twelve function keys of the PC, alone and in combination with Ctrl, Alt, and Shift keys.

Menucolors Set the colors in the menu environment.

Mode Define user mode, either Safe or Expert. In Safe mode, you cannot make functional changes to the design.

Mouse Set the mouse-connection port and the mouse-button definitions.

Namechange Update the current design file name (on/off switch).

Palette Control the colors that XDE uses.

Part Control the part type for the current design.

Printer Control the current printer type.

Speed Control the part speed.

You can change these options using XDE Executive commands. The changes remain in effect until you leave the XDE program. To make permanent changes, use the Saveprofile command. The current settings, except for directory, are saved in the xact.pro file. The next time you start XDE, the program reads and loads the settings from the xact.pro file.

The current settings for some of these options appear on the display, as mentioned above. Use the Settings command to see the current settings for all the options.

Leaving the XDE Program

The Quit command leaves the XDE Executive and returns to the operating system. If you have made changes to a design file but have not saved those changes, a prompt asks if you want to save them.

The File command saves the current state of the design file, leaves the XDE Executive, and returns to the operating system.

The DOS command suspends operation of the XDE program and exits to the operating system temporarily. Typing Exit at the operating system prompt returns you to the XDE Executive, to the state it was in when you entered the DOS command, which is available only on PCs.

NOTE

You must save changes made to a design file by using File or Save before leaving the XDE program; otherwise, the changes will be lost.

XDE Executive Commands

This section lists each command alphabetically and explains its use. The command description includes the menu which contains the command, the keyboard syntax, and any keyboard abbreviations.

Beep — Control Beep Sound

Menu	Profile
Syntax	beep on|off
Abbreviations	none

Beep is a toggle that controls whether the beep is on or off.

Convert — LCA Part Type Conversion Program

Menu	Programs
Syntax	convert *filename part_type*
Abbreviations	co

Convert automatically changes the current LCA device to another LCA part type. You can only use this command to change a smaller part to a bigger part within the same family. For instance, you can automatically convert a design for the 2000-gate XC3020 in a 68 PLCC package (part 3020PC68) into a design for the 3000-gate XC3030 in an 84 PLCC package (part 3030PC84).

Start the Design Rules Checker (DRC) after running Convert to ensure that all nets are routed and that the net delays in the new design are acceptable.

During conversion from an XC3020 (a smaller part within a family) to an XC3030 (a larger part within a family), the XC3020 design can be mapped onto the upper left-hand or lower right-hand corner of the XC3030. If you select the lower right-hand corner, the routing of all non-I/O signals remains the same. If you select the upper left–hand corner, signals that pass through bidirectional buffers are rerouted. CLBs in the converted source design map directly to the corresponding CLBs in the destination part.

IOBs in the source design map directly to corresponding IOBs in the destination part where possible. Convert generally reroutes the nets from internal signals to the IOBs.

It is possible to convert a part type to a different package while retaining CLB placement and routing. However, if the destination package has fewer bonded IOBs, some IOBs might be unbonded. If a design is converted to a package type with fewer bonded pins, for example, from a 84-pin 3020PC84 part to a 68-pin 3020PC68 part, several pins in the destination design are unbonded because the destination IOB is one of those on the die that is not bonded. You are responsible for moving IOBs that end up on unbonded pins unless it does not matter whether the IOB is bonded to a particular package pin. Unbonded IOBs, if not named in the source design, are labeled U*xx* where *xx* is unique to each IOB.

You cannot convert a design from a larger die to a smaller die, for example, XC3030 to XC3020. To convert a larger die to a smaller die, use the LCA2XNF program to generate an XNF file, and then run XNFMAP and MAP2LCA to generate the LCA file. You can also use the XDE Cut and Paste commands to extract portions of the source design into files that you can load into the destination design.

You cannot convert designs between different architectures; for example, between XC2000 and XC3000 families. If you entered the design with schematic entry, you might move the design to another LCA family if unique features of the original device are not used, such as 5-input functions on an XC3000-family design or CLB latches on an XC2000 design.

Cursor — Set the Cursor Shape

Menu	Profile
Syntax	cursor *filename*
Abbreviations	c

There are four cursor shape options.

Arrow	A slanted arrow whose tip is the cursor location
Bug	A small X whose center is the cursor location
Cross	A large cross whose center is the cursor location
Gunsight	A screen-width cross centered at the cursor location (PC only). On the workstation, it is identical to the cross.

You can abbreviate these options to A, B, C, and G, respectively. The xact.pro file contains the initial setting for this option. The default setting is Arrow.

Design — Set the Current Design File

Menu	Designs
Syntax	design *[new] filename*
Abbreviations	de

Design specifies the name of the design file submitted to the XDE XDE Executive. If you omit the New option, the file must already exist. All XDE programs except for the Design Editor operate only on an existing design file.

The specified file name becomes the default file name that the Save and File commands use. Running either the Save or File commands creates the design file.

Directory — Set the Current Directory

Menu	Designs
Syntax	directory *directory*
Abbreviations	di

Directory sets the current directory to which the XDE program has immediate access. The current directory name is shown on the XDE Executive display.

You must enter the directory name in PC DOS format on a PC. In the XDE program, both the forward slash (/) and the DOS standard backslash (\ ) are valid directory separators. Here are two examples:

```
Cmd: dir C:\Xilinx\Designs

Cmd: dir C:/Xilinx/Designs
```

DOS — Enter Temporary DOS Shell (PC only)

Menu	Programs
Syntax	dos *DOS_command*
Abbreviations	do

DOS allows you to execute a standard DOS function, which is not available from the XDE program, without quitting the current XDE session.

If you do not specify a DOS command, the operating system prompt appears. Typing Exit brings back the XDE Executive in the state it was in when you entered the DOS command.

If you specify a DOS command, it is executed immediately; then the XDE XDE Executive resumes.

When operating DOS, available memory is limited. Some DOS commands might not operate correctly and result in an error message because of the amount of memory they require. Running DOS commands that install programs in memory can affect subsequent XDE operations that attempt to allocate memory.

DRC — Start the Design Rules Checker

Menu	Programs
Syntax	drc *filename options...*
Abbreviations	none

DRC checks the specified design file for violations of design rules such as multiple sources on a net. If the file is not currently in memory, XDE loads the file.

EditLCA — Start the Design Editor

Menu	Programs
Syntax	editlca *[new] filename*
Abbreviations	e

EditLCA provides the functions to create new design files and alter existing ones. It can manipulate interconnect and define the logic of CLBs and IOBs. A completed design file is used to generate the bitstream to configure an LCA device.

NOTE *With the introduction of XDE 4.1, a Safe mode is available. In this mode, you cannot make logic changes to the design file in the XACT Design Editor. To make such changes, set the mode to Expert.*

Start EditLCA on the current design file. If the file is not currently in memory, it is loaded. The Editor subprogram begins operating, displaying the Physical Interconnect Editor (PIE) screen.

EditLCA generates a DBK file whenever you crash or break out of the program. It generates a LOG file whenever you exit the program without first saving your edits. Finally, EditLCA generates an ODF file and an OLF file when you save your edits.

Execute — Execute a File of Commands

Menu	Profile
Syntax	execute *filename*
Abbreviations	ex

Execute executes a file of commands. The specified file name is assumed to be text file that contains EditLCA and/or XDE Executive commands. Execute performs the commands sequentially as they appear in the file.

Fastsave — Enable/Disable Fast Design File Read/Write Operations (PC only)

Menu	Profile
Syntax	fastsave on I off
Abbreviations	fasts

Fastsave enables or disables the fast writing and reading of design information by the XDE program. When Fastsave is enabled and you run File or Save, the XDE program creates three files. One is the normal LCA file; the other two files are binary files whose contents can be quickly read and written. The three files and their contents are described below.

• ***name***.lca	Normal LCA design file
• ***name***.xgr	Binary graphics file
• ***name***.xcd	Binary LCA design file

When Fastsave is enabled and you execute a command that causes a design to be read, the XCD and XGR files are read if they exist and are newer than the LCA file. When you execute Save or File, the LCA file is always written first. The dates on the two files and other information ensures that the XCD and XGR files are always properly locked to the LCA file.

The XCD and XGR file formats are subject to change across revisions of the XDE program. Do not regard either of these files as the ultimate design record. The two files are intended to provide fast cycles into and out of the XDE program.

Reading the two binary files should take roughly 10 or 20% of the time it takes to read the corresponding LCA file. Reading the binary files is also independent of the fullness of the LCA design, where reading an LCA file takes longer as the file gets larger.

The two binary files can be quite large; for an XC3090 design, each is roughly 1.2 MB, for a total of 2.5 MB.

When Fastsave is disabled, XDE creates only the LCA file when Save or File is executed. When Fastsave is disabled and a design is read, the XDE program looks only for the LCA file, even if the XCD and XGR files exist.

File — Save the Current Design and Leave XDE

Menu	Designs
Syntax	file *file name*
Abbreviations	f

File saves the current design in the specified design file and takes you one level back. For example, if you are in the XDE Executive and you started XDE from the operating system, it takes you back to the operating system prompt. If you started XDE from XDM, it takes you back to the XDM opening screen. If you are in the EditLCA menu, executing File saves the current design in the specified design file and returns you to the XDE Executive. If you do not specify a file, XDE uses the current file name. The XDE program automatically adds the LCA extension to the file name, if you do not specify an extension.

Use Save to save the current design without leaving the XDE program. Use Quit to leave XDE without saving the current design.

If the design cannot be saved, XDE issues a message and the current session continues.

Help — On–line Explanation of Editor Commands

Menu	Profile
Syntax	help
Abbreviations	none

The Help command displays a menu of all the topics for which on-line help is available. While this menu is open, use the mouse to select a topic to get more information.

Help is also available at any time by pressing F1 when the mouse is over a menu item, or in the EditLCA display. When the mouse is in the EditLCA screen, pressing F1 displays information about the LCA object under the cursor.

Keycursor — Activate Cursor Keys for Pull-Down Menus

<table>
<tr><td>Menu</td><td>Profile</td></tr>
<tr><td>Syntax</td><td>keycursor on|off</td></tr>
<tr><td>Abbreviations</td><td>keyc</td></tr>
</table>

When Keycursor is off you must enter commands with the keyboard or use the mouse to select commands from pull-down menus. With this option on, the arrow keys move the cursor through pull-down menus; pressing ↵ executes the selected option.

Some commands present menus of blocks or nets that require more than one screen to display. For these menus, you can use the PgUp and PgDn keys to move between pages, and the Home and End keys to move to the beginning or end of the list.

Keydef — Define A Function Key

Menu	Profile
Syntax	keydef *keyname_definition*
Abbreviations	keyd

The key name must be one of the twelve function keys F1 through F12. You can also combine any function key with the Shift, Ctrl, or Alt key by holding them down at the same time as the function key.

The keystrokes you type as the function key definition are stored under that function key; whenever you press the function key, the stored commands are executed. Use function keys to store command sequences you use frequently.

You can terminate a function key definition by pressing Enter or the backslash (\ ). If you terminate the keystroke sequence by pressing Enter, the carriage return is included in the key definition. If you terminate a key definition, the command defined by the key displays and executes immediately when you press the key. If the definition is terminated with a backslash (\ ), the key definition displays, but does not execute when you press the key. You must press Enter to execute commands so defined.

To delete a function key definition, enter Keydef and the key name, then press Enter without typing a definition. The initial settings for this option are contained in the xact.pro file. These function key definitions are for the XDE Executive only. Each subprogram has its own set of function key settings.

The F1 key is reserved for on-line Help and cannot be redefined. However you can still use it in conjunction with the Shift, Ctrl, and Alt keys.

MakeBits — Start the Bitstream Generator Program

Menu	Programs
Syntax	makebits *file name*
Abbreviations	makeb

MakeBits, the bitstream generator, creates a configuration bitstream file for the specified design file. It provides many options, including the option to tie down unused interconnect and examine the effects. In addition, the Download cable software is included in the MakeBits program. See "The MakeBits Program" chapter in Volume 2 of the *XACT Reference Guide* for more details.

MakePROM — Start the Prom File Formatter

Menu	Programs
Syntax	makeprom
Abbreviations	makep

The PROM file formatter creates and saves PROM memory images. It provides options to create several different PROM file formats.

Each PROM file can contain multiple BIT files, as well as daisy chains of BIT files. You can use the PROM file to program a Xilinx Serial Configuration PROM or other memory devices. The default PROM size is 64K. See "The MakePROM Program" chapter in Volume 2 of the *XACT Reference Guide* for more details.

Menucolors — Modify XDE Executive Screen Colors

Menu	Profile
Syntax	menucolor *option color*
Abbreviations	men

Menucolors selects a particular item in the XDE Executive screen, such as Menuitemcolor, Cmdtextcolor, and Cursorcolor, and defines its color. The Reset option returns all the colors to their default value. To save the changes, use the Saveprofile command.

Mode — Set the Mode to Safe or Expert

Menu	Programs
Syntax	mode
Abbreviations	mod

With the introduction of XDE 4.1, a Safe mode is available. The Safe mode does not permit you to make logic changes in the XDE. To make changes, set the mode to Expert.

In the Safe mode, commands that change the functionality of your design, such as Addnet, Delpin, and any commands within the Config Menu are not available. Safe mode guards against accidental changes to the logic in your design at the LCA level. In Expert mode, you have full access to all the XDE commands.

If you are in Safe mode, and you try to change some logic within the design, XDE reports that the command you are trying to use is unknown. If this happens, and you want to change logic, you must go back to the XDE Executive and change the mode from Safe to Expert.

Mouse — Set the Mouse Configuration

Menu	Profile
Syntax	mouse *option function*
Abbreviations	m

Mouse performs two operations.

- It defines the PC-communications port used by the mouse.
- It defines the mouse-button functions.

Mouse-port definition is set by one of two options.

- COM1 defines the mouse port as the PC COM1 port.
- COM2 defines the mouse port as the PC COM2 port.

You determine mouse button functions by selecting one of three options.

- B1 selects mouse button #1
- B2 selects mouse button #2
- B3 selects mouse button #3. It is valid only if a third button exists.

If you select B1, B2, or B3, you must also enter one-of-four functions that determine the action of the selected button.

- Select enters the cursor location. Normally, it choose an item, either in a menu or in a display.
- Done enters the Done command. Normally, it executes the current command and continue the XDE session.

- Menu displays the most recently displayed menu and moves the cursor to the most recently selected item. If a menu is already displayed, the mouse button enters the cursor location (same as Select).
- Switch switches between the PIE display and the Block Editor display of the current block. Switch is useful only in the Design Editor program.

The initial setting for the Mouse option is in the xact.pro file. The Design Editor has its own setting for the mouse configuration; therefore, the mouse configuration can change when you start the Editor.

Namechange — Change the Current Design File Name

Menu	Profile
Syntax	namechange on|off
Abbreviations	namec
See also	Save, File

If Namechange is on, the file name used in a Save or File command becomes the new default file name and the XDE Executive Screen is updated to reflect this change. With Namechange turned off, the current file name is always retained. In this case, when you use Save or File with a new file name, the file is written to the new file name, but the original file name is still the default.

Palette — Select Available Color Palettes

Menu	Profile
Syntax	palette *name*
Abbreviations	pal

The Palette command allows you to select a color palette.

Part — Set the LCA Part to be Configured

Menu	Designs
Syntax	part *partname*
Abbreviations	par

The Part command determines the LCA type and package type to be used in new designs. If you do not specify the optional part name, you can select from a menu of supported parts.

Printer — Set the Printer Type

Menu	Profile
Syntax	printer *type*
Abbreviations	printe
See also	PR

The printer type determines the format that the Print command uses to create the printable file. Table 3-4 lists the valid printer types and associated formats. You can change the initial value for the printer type in the xact.pro or editlca.pro file; the default value is IBMgraph.

Table 3-4 List of Printers and Print Formats

Model	Make
FX80	Epson FX80
FX85	Epson MX85
FX86	Epson MX86
FX100	Epson FX100
FX185	Epson FX185
FX286	Epson FX286
HPLASER	Hewlett Packard Laserjet Printer
HPLPLUS	Hewlett Packard Laserjet Plus Printer
HPLPLUSLEGAL	Hewlett Packard Laserjet Plus
IBMgraph	IBM Graphics Printer
MX80	Epson MX80
MX100	Epson MX100
NEC	Nippon Electric Corp.
NECBIG	Nippon Electric Corp. (expanded)
OKI92	Okidata Microline 92
OKI93	Okidata Microline 93
OKI292	Okidata Microline 292
OKI293	Okidata Microline 293
POSTSCRIPT	Postscript Format Printer
RX80	Epson RX80

Printpic — Print a PIC File

Menu	Programs
Syntax	printpic PIC_*file*
Abbreviations	printp

This command sends the named PIC file to your printer. It prints out only PIC files and not PMF files, so you can use this command at any time from within the XDE program. It performs the equivalent of the following DOS command.

```
copy /b file name prn:
```

Quit — Leave XDE

Menu	Programs
Syntax	quit yes|no
Abbreviations	none

The Quit command ends the XDE session. If you have made changes to a design file but have not saved those changes, you are prompted with `ARE YOU SURE?` when you enter the Quit command. This prompt allows you to cancel the Quit command to save the changes. If you do not save the changes before leaving the XDE program, they are lost.

Read — Load a Design File into Memory

Menu	Designs
Syntax	read *file name*
Abbreviations	r

The Read command loads a specified design file into memory. If you do not specify a design file, Read loads the design data in the current design file into memory. If there is already design data in memory, it is replaced by the loaded data. Once the data has been loaded, XDE subprograms can operate on the design. If there is no design data loaded when you start an XDE subprogram, the Read command is automatically performed before the subprogram is run.

Readpalette — Read the palettes.xct File

Menu	Profile
Syntax	readpalette
Abbreviations	readpa

If you modified the palettes.xct file (with a text editor), you can use this command to re-read it without having to exit the XDE program and restart. After you read the palettes.xct file, you can choose a palette with the Palette command.

Readprofile — Load XDE Executive Option Settings from xact.pro

Menu	Profile
Syntax	readprofile
Abbreviations	readpr
See also	Saveprofile

The xact.pro file contains settings for the following options: Cursor, Directory, Design, Fastsave, Keycursor, Keydef, Menucolor, Mouse, Namechange, Palette, Part, Printer, and Speed.

The Readprofile command sets these options to the xact.pro file values. This command is useful for setting XDE Executive options to known states.

When you start the XDE program, the default xact.pro file is loaded. If none exists, the xact.pro file located in the XDE program is loaded. To gain access to another xact.pro file, use the Directory command to switch to the desired directory, and then start the Readprofile command.

Save — Save Current State of the Design

Menu	Designs
Syntax	save *filename*
Abbreviations	sa

This command saves the current state of the design in the specified design file. XDE automatically adds the LCA extension to the file name if you do not. If you do not specify a file, Save uses the current file. When the design has been saved, XDE resumes.

Saveprofile — Save XDE Executive Option Settings in xact.pro

Menu	Profile
Syntax	saveprofile
Abbreviations	savep

This command saves the XDE option settings in the xact.pro file. The xact.pro file contains settings for the following options: Cursor, Directory, Design, Fastsave, Keycursor, Keydef, Menucolor, Mouse, Namechange, Palette, Part, Printer, and Speed.

The Saveprofile command sets the xact.pro file values to the current settings. Use this command to save the XDE Executive option settings for subsequent XDE sessions.

Settings — Display XDE Executive Option Settings

Menu	Profile
Syntax	settings
Abbreviations	se

This command displays the current settings for the following options: Cursor, Directory, Design, Fastsave, Keycursor, Keydef, Menucolor, Mouse, Namechange, Palette, Part, Printer, Speed.

The display includes only those function keys that you have defined.

Speed — Select Device Speed Grade

Menu	Designs
Syntax	speed *speedgrade*
Abbreviations	sp

This command sets the device speed grade for the current design. The speed grade is used by the Delay, Autotime, and Querynet functions in EditLCA for determining signal propagation delays.

XDE Printer Support

This section provides information about using printers that the XDE program does not support. It also describes the printcap.xct file and provides information about the printer commands you can use with the printer name.

Unsupported Printers

The Printer command in the XDE Executive and the EditLCA program permits selection of one of the printers supported by the XDE program. Designers using unsupported printers can modify a printer description text file (printcap.xct) to accommodate their printer to the printer list for use with the Print command. This file is in the \XACT\DATA directory.

Since future updates to the XDE program can overwrite this printcap.xct file, you should keep a separate copy of your own printer additions as a backup, so that the same enhancements, if required, can be added to future versions of printcap.xct.

Printcap.xct File

Each printer description consists of two parts: the printer name as it will appear in the Printer menu and the printer escape-sequence commands to perform various actions.

The printer name starts in the first column. Multiple printer names on separate lines, each starting in the first column, indicate that several different printers respond to exactly the same escape sequences.

Printer Commands

Immediately following the printer names, include one or more pairs of printer commands and values. Each command is preceded by white space consisting of at least one space or tab character. Table 3-6 lists the XDE printer commands. A *value* should follow the command on the same line, separated by one or more spaces and tabs. There are three general rules for the *values* format.

- There can be no embedded white space.
- Numeric quantities must be in decimal.
- Non-ASCII characters in escape sequences can be represented by the following ASCII character sequences.

Table 3-5 Escape Sequences

\E	ESC character
\t	tab character
\n	newline character
\r	carriage return character
\f	form feed character
\\	backslash character
\nnn	any character, where *nnn* is a three-digit octal number

Table 3-6 XDE Printer Commands

Command	Description
COLOR1 *value*	The escape sequence selects GREEN.
COLOR2 *value*	The escape sequence selects RED.
COLOR3 *value*	The escape sequence selects YELLOW.
CR *value*	This is the escape sequence that causes a graphic carriage return.
EG *value*	This is the escape sequence that causes the printer to enter graphic mode.
EP *value*	This is the escape sequence given at the end of a page.
END *value*	This is the escape sequence given to exit graphic mode.
LSB	This indicates that the top row of dots in the printhead is mapped into the least significant bit of the data byte.
MSB	This indicates that the top row of dots in the printhead is mapped into the most significant bit of the data byte.
NL *value*	This is the escape sequence that causes a graphic newline.
PINS *value*	This indicates the number of pins in the printhead.
SL *value*	This is the escape sequence given at the start of a line.
WIDTH *value*	This indicates the paper width in pixels.
HEIGHT *value*	This indicates the paper height in pixels.
OKI92	This is an Okidata 92 type printer that needs special handling for certain graphics data values.
LASER	This is a laser printer that processes the graphics data horizontally rather than vertically, like a dot matrix printer.
INIT *value*	This is the escape sequence that should be used to initialize the printer.

SL (Start of Line) Command

In addition to the general rules for specifying values for commands, there are special rules for the SL command. Many printers require the computer to specify the line length (bytes of graphic information) at the beginning of each line. The format of this information is printer-dependent; accordingly, two formats are provided for telling the XDE program how to place this number into the escape sequence.

```
%i
```

Place the byte count in binary into the next two characters with the least significant byte first and the most significant byte last.

%*number* d

Place the byte count in ASCII decimal characters into the next number characters. If number starts with a zero, fill the most significant digits with zeros.

Each of these options is illustrated in the SL command in one of the sample printcap.xct entries below.

Examples

Sample printcap.xct

Three sample entries are shown below. Refer to your printcap.xct file (in the \XACT\DATA directory on PCs, or in $XACT/data on workstations) for additional samples. You can include comments after the value entry, separated from the value entry by white space.

FX80

The entries for the FX80 printer are listed in Table 3-7.

Table 3-7 FX80 Printer Entries

ep	\f	At termination issue, FORMFEED
eg	\E3\030	Set line spacing to 24/216”
sl	\EK%i	<ESC>K<n1><n0> on each line
nl	\n	At end of line, issue NEWLInE
cr	\r	And CR
pins	8	Printer has 8-pin print head
width	480	Printer is 480 dots wide
msb		First row of pins is the MSB

HPLASER

The entries for the HP LaserJet printer are listed in Table 3-8.

Table 3-8 HP LaserJet Printer Entries

laser		Printer is a laser type (one pin)
iinit	\EE\E&l2E\E&a0V	Initialize the printer
eg	\E*t75R\	Enter graphics in 75 DPI resolution
width	600	Printer is 8×75 dots wide
height	710	Printer is 9.47×75 dots high
ep\	E*rB\f	Exit graphics, FORMFEED is at page end
sl	\E*b%dW	Print on each line
end\	E*rB\EE	

POSTSCRIPT

The entries for the PostScript printer are listed in Table 3-9.

Table 3-9 PostScript Printer Entries

top_margin	1.0
left_margin	1.0
right_margin	1.0
bottom_margin	1.0
printer_resolution	300
title_font	Helvetica
print_titles	
top_overlap	0
right_overlap	0
fit_to_page	
page_height	11.00
page_width	8.500
scale_factor	3

Design Editor Display

The Design Editor is a graphics program that displays and manipulates the functional layout of an LCA (Logic Cell Array). The Design Editor operates on the XC2000, XC3000, or XC4000 families of LCAs as selected from the XDE XDE Executive. To learn more about the internal composition of an LCA, see the *Xilinx Programmable Logic Data Book.*

The Design Editor consists of the following graphic displays:

- Physical Interconnect Editor (PIE) is a graphic display for configuring the interconnect locations in the LCA device
- Block Editor is a graphic display for configuring the internal logic of CLBs (Configurable Logic Blocks) and IOBs (Input/Output Blocks) and other configurable blocks, such as the boundary-scan block in the XC4000

The PIE operates on pins, nets, and blocks. Pins are the inputs and outputs of CLBs and IOBs; they are internal points and should not be confused with the LCA external package pins. A net is a collection of connected pins. The PIE includes an automatic router that chooses the routing (interconnect) for a net. You can alter or eliminate this routing without changing the net pins. Blocks are either CLBs or IOBs. You can move blocks to different locations, define their pin connectivity, and alter their configuration information.

The Block Editor operates on one block at a time. When you select a block for editing, that block becomes the current block, and the Block Editor display appears. All Block Editor commands are performed on the current block.

To start the Design Editor from the XDE program, enter the EditLCA command at the XDE Executive prompt, or select EditLCA from the Programs menu.

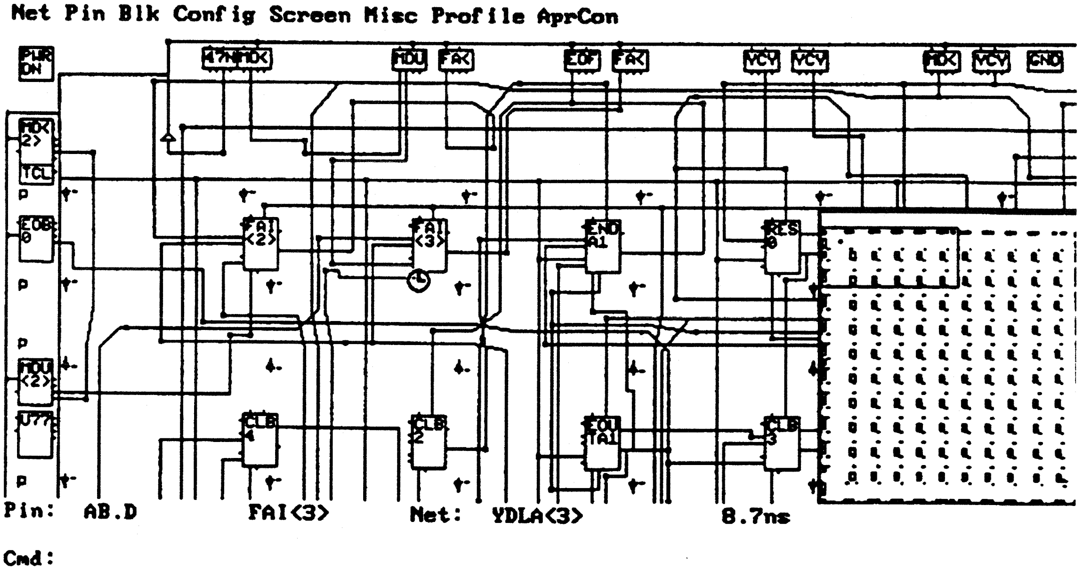
Physical Interconnect Editor Display

The PIE display appears when you start the Design Editor from the XDE program and consists of the following:

- World view (an optional map of the entire LCA)
- LCA layout
- Cursor
- Cursor status line
- Pull-down menus
- Message line
- Command line

World View

Figure 3-11 shows the PIE display. The rectangular diagram overlaying the lower right corner is the world-view, showing all blocks in the LCA device. The interconnectable LCA blocks are shown behind the world-view diagram in a full-screen display. The blocks that are visible in the full-screen display are shown within the world-view red window. To choose other areas of the LCA device, you can move the window by holding down any button on the mouse while dragging it to the new area. The Design Editor includes commands that alter the window size and turn the world-view display on and off. The Show command, under the Screen menu, sets these options.



```
Pin:  AB.D          FAI<3>      Net:  YDLA<3>          8.7ns

Cmd:
```

Figure 3-11 Physical Interconnect Editor Display

LCA Layout

Each CLB in the PIE display is initially identified by its row and column letters (XC2000 and XC3000), or row and column numbers (XC4000). Each IOB is identified by its LCA package pin number. CLB and IOB pins (inputs and outputs) are drawn as shown in Figure 3-12, Figure 3-13, and Figure 3-14. Each pin is named according to its block and input or output name. For example, HA.C is the C input of the CLB in row H and column A (XC2000 and XC3000); P24.T is the 3-state control input of the IOB at LCA package pin 24.

Display commands are available in the Screen menu to enable or disable the display of interconnect details in several scales. Programmable Interconnect Points (PIPs) are shown as tiny squares; switching matrices are shown as larger squares having several pins on some or all sides (see Figure 3-15 through Figure 3-17).

Turn on or off the display of PIPs, switching matrices, splitter PIPs, and unconfigured direct interconnects by selecting the appropriate option under the Show command. An initial value set in the editlca.pro file determines whether they are initially displayed when the Editor is started. You can select and save Profile options to modify the profile.

The global clock buffers and oscillators are always displayed on the die. Reserved package pins are also always displayed.

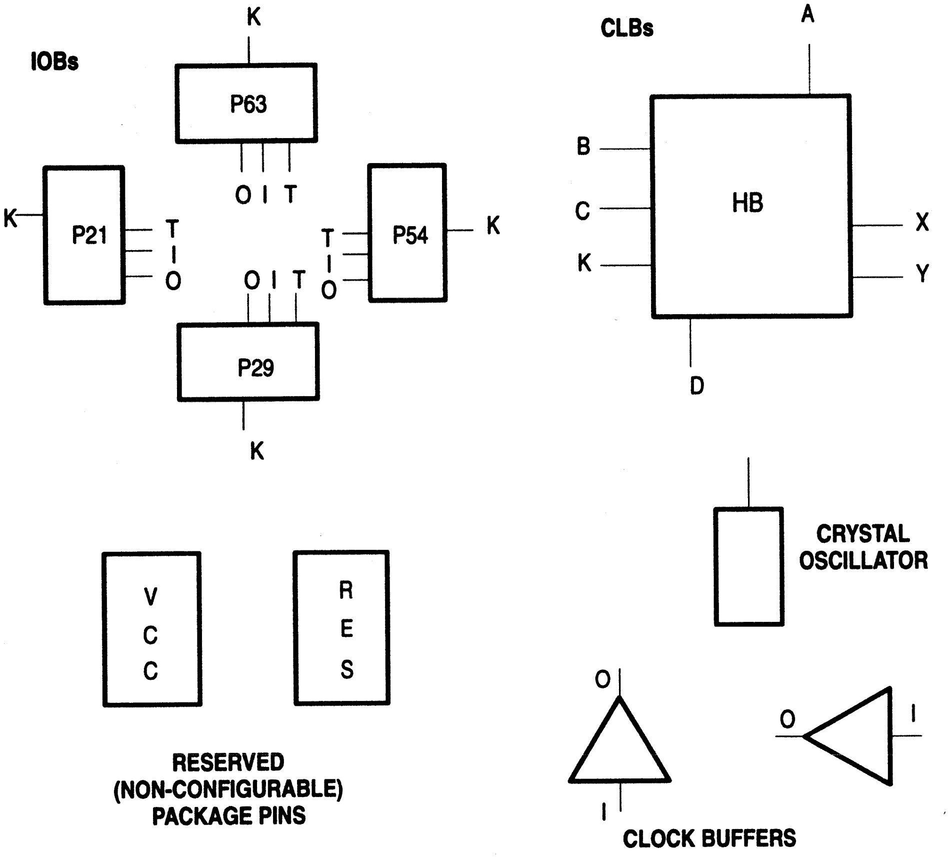


Figure 3-12 Block Inputs and Outputs (XC2000)

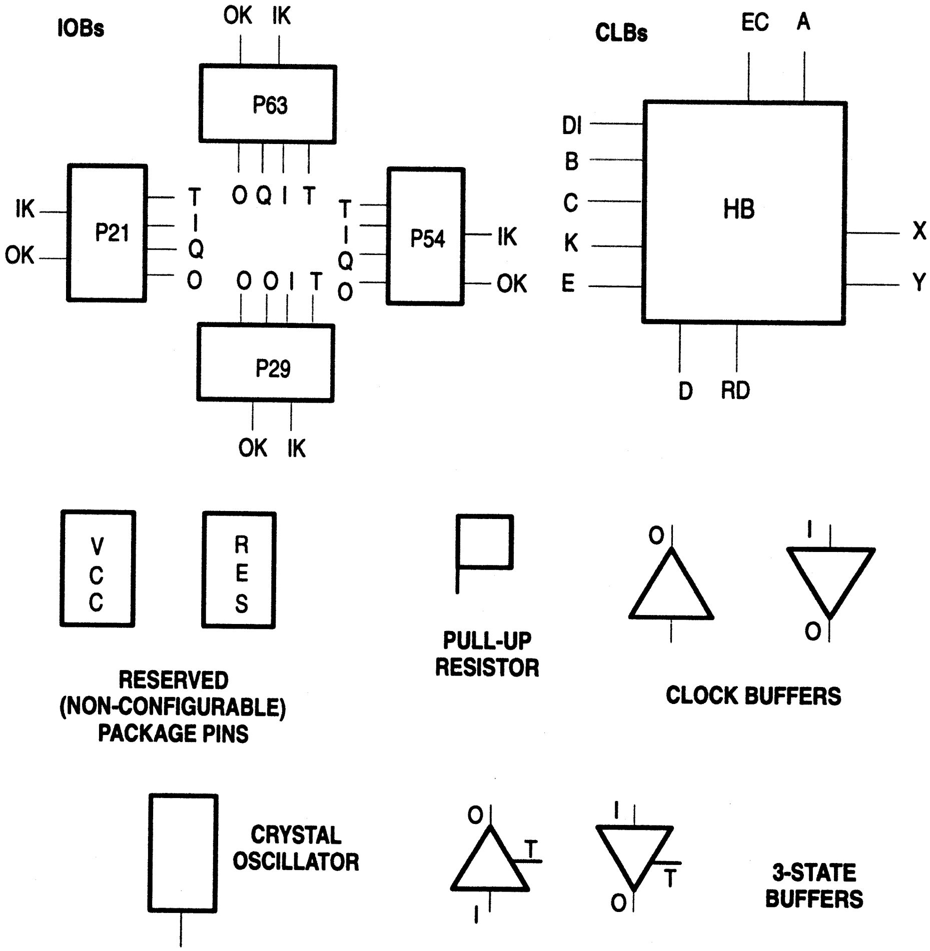


Figure 3-13 Block Inputs and Outputs (XC3000)

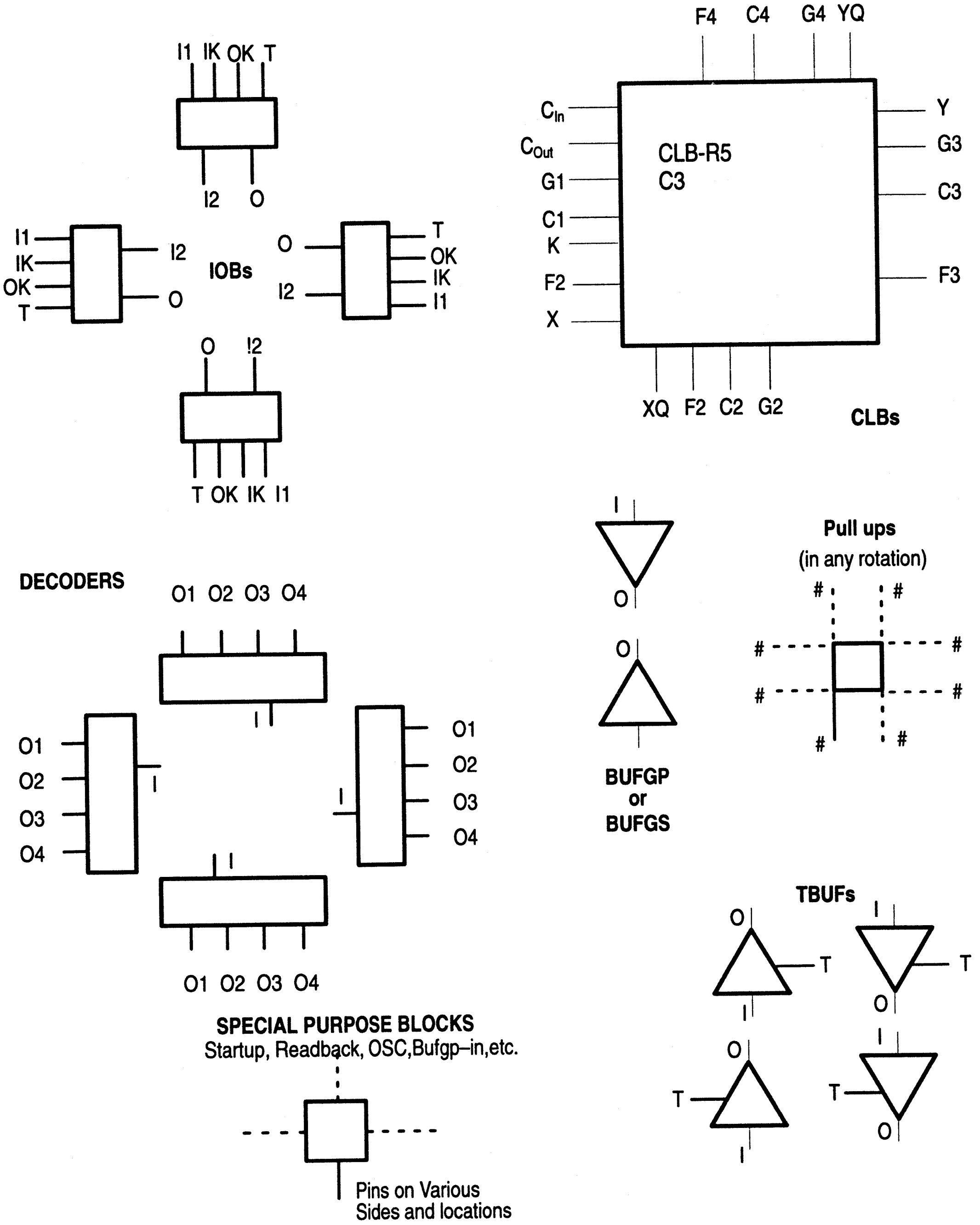


Figure 3-14 Block Inputs and Outputs (XC4000)

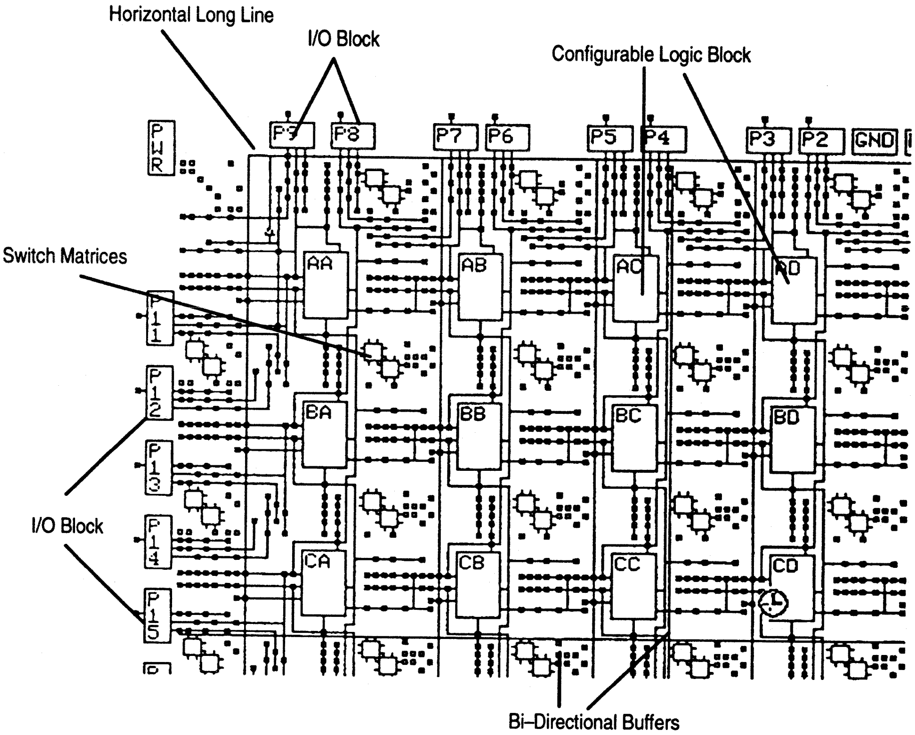


Figure 3-15 LCA Layout Elements (XC2000)

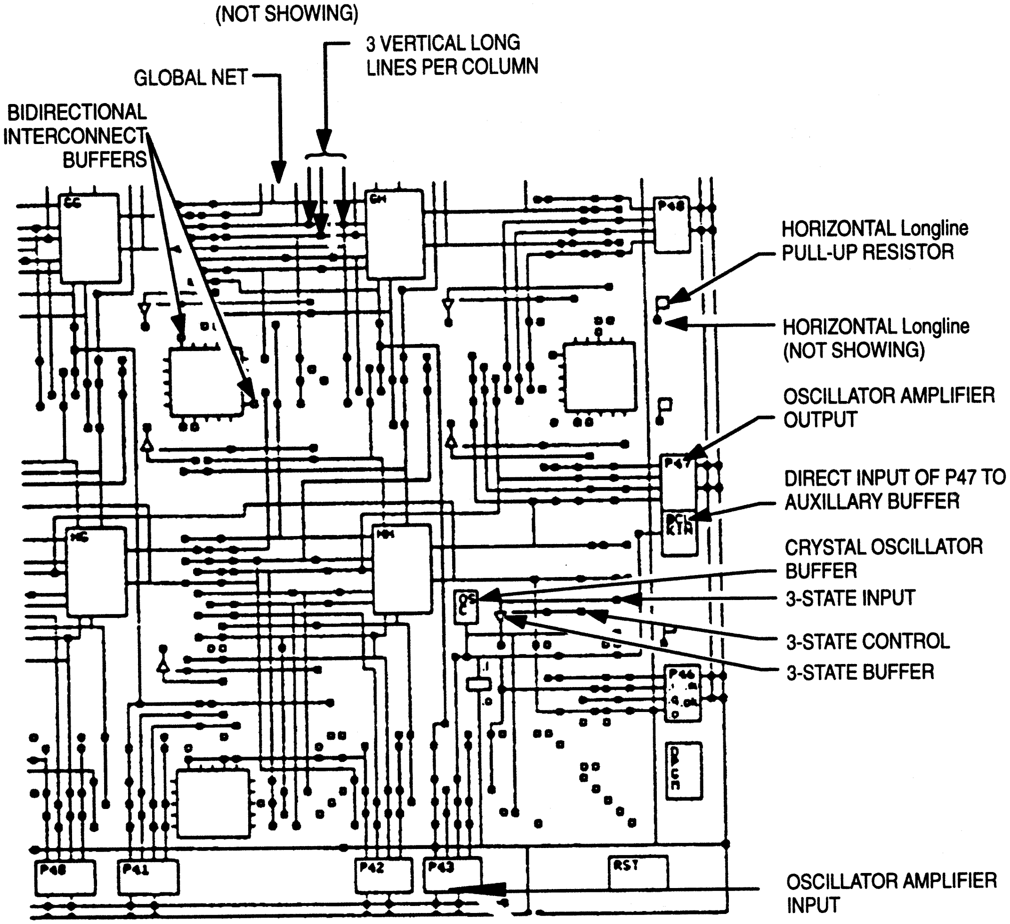

(NOT SHOWING)
3 VERTICAL LONG LINES PER COLUMN
GLOBAL NET
BIDIRECTIONAL INTERCONNECT BUFFERS
HORIZONTAL Longline PULL-UP RESISTOR
HORIZONTAL Longline (NOT SHOWING)
OSCILLATOR AMPLIFIER OUTPUT
DIRECT INPUT OF P47 TO AUXILLARY BUFFER
CRYSTAL OSCILLATOR BUFFER
3-STATE INPUT
3-STATE CONTROL
3-STATE BUFFER
OSCILLATOR AMPLIFIER INPUT
GG
GH
P48
P47
HG
HH
OS
P44
P40
P41
P42
P43
RST

Figure 3-16 LCA Layout Elements (XC3000)

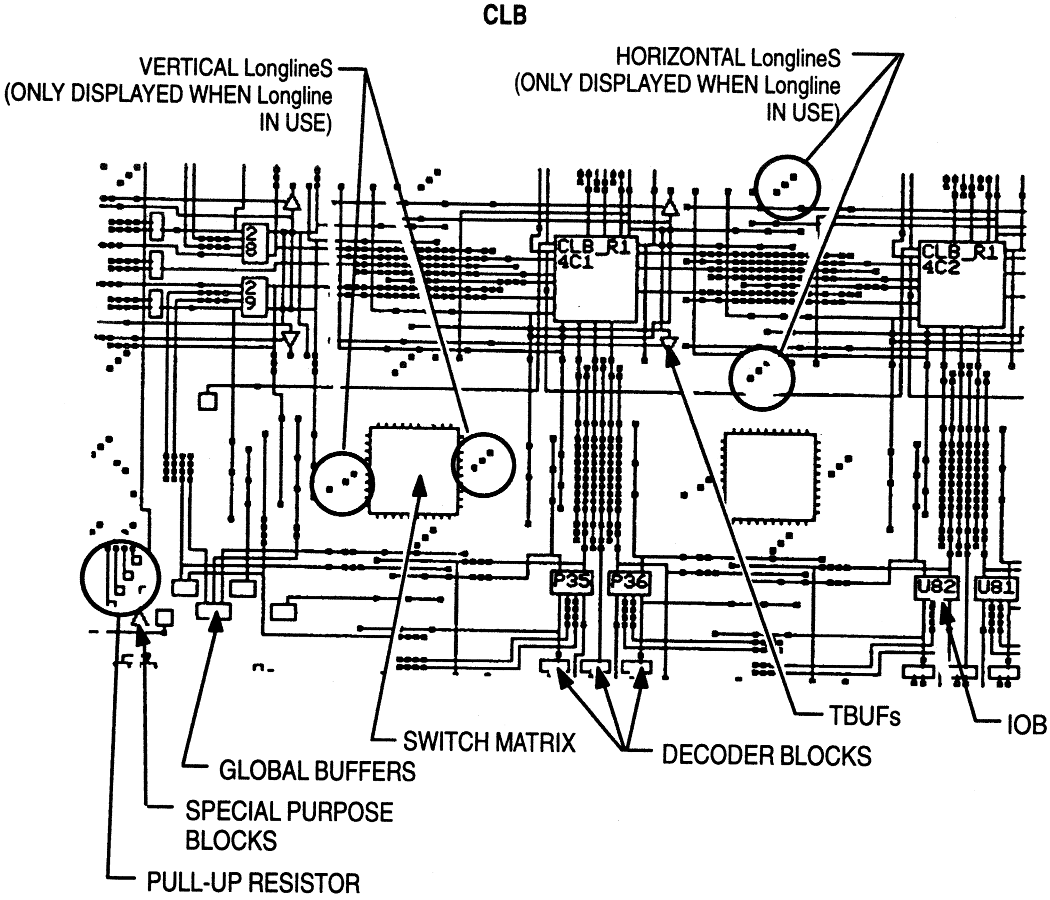


Figure 3-17 LCA Layout Elements (XC4000)

Cursor

The initial cursor shape in the Design Editor is set in the XDE Executive or in the xact.pro or editlca.pro files. You can change the cursor shape any time using the Cursor command. In addition to selecting commands and menus with the cursor, you can also enter locations. If a particular command requests a location, pin, net, or a block, you can point the cursor at one of these items and select it using a mouse button, instead of using the keyboard.

Cursor-Status Line

The cursor-status line gives information about the location of the cursor. This information can be a pin or block name, a PIP, or a switching-matrix pin. If it is a pin name, the cursor-status line also displays the name of the net, if any, to which the pin belongs. As you move the cursor around the display using the mouse, the information on this line changes to show the exact item identified by the cursor position. If the Autotime option is on, the delay associated with each load pin is also displayed.

Pull-Down Menus

The names of at least nine pull-down menus appear across the top of the display: Net, Pin, Blk, Config, Screen, Misc, Profile, Aprcon (for XC2000, XC3000 devices only), Probe, and Timing.

Message Line

The message line displays messages that signal errors, indicate the completion of a function, or prompt for information. If the message is more than one line, it appears on the text display instead of the Design Editor graphic display. In a single monitor system, the text display temporarily replaces the Editor display. Pressing any key displays the next page of text, if any, on the screen. After all the text has been displayed, press any key to return to the Editor display.

Command Line

The command line displays a prompt that indicates the type of information the program needs. When you begin the program, the prompt reads `Cmd:`, indicating that any valid command is an appropriate input. At other times, the prompt might request other input, such as a pin or block name.

Block Editor Display

When you select a block for editing, the PIE display is replaced by one of the Block Editor displays shown in Figure 3-18 through Figure 3-20, depending on whether the selected block is an IOB or CLB. You can switch between the Block Editor display and the PIE display.

The Block Editor display consists of the following:

- Block configuration options
- Block configuration diagram
- Pull down menus
- Cursor
- Cursor status line
- Message line
- Command line

- Block identification
- Net assignment
- Net/tag association

The menus, cursor, cursor status line, message line, and input line are the same as the PIE display.

The following legend identifies the regions of block editor display for the next three figures. Not every block display contains all regions.

1 Schematic drawing region
2 Net binding region (maps pins to nets)
3 Function generator equations region
4 Karnaugh map region (1 per function generator)
5 Truth table region (1 per function generator)
6 Pin binding region (maps pins to configuration tabs)
7 Block configuration tags region
8 Miscellaneous region (contents depend on block type)

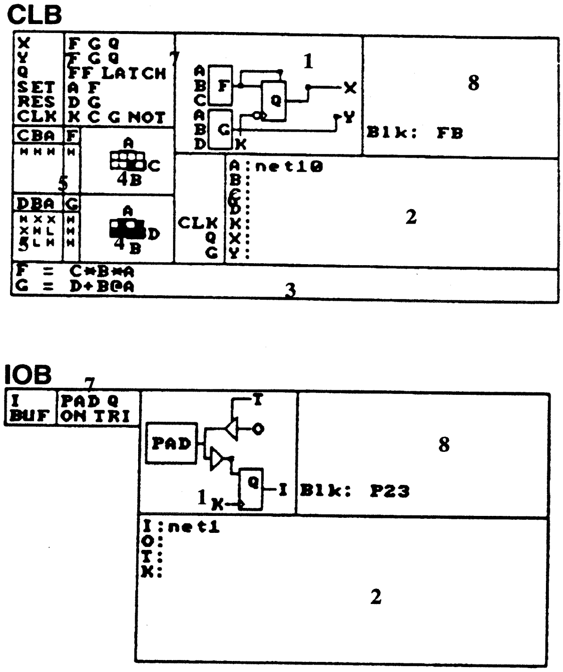
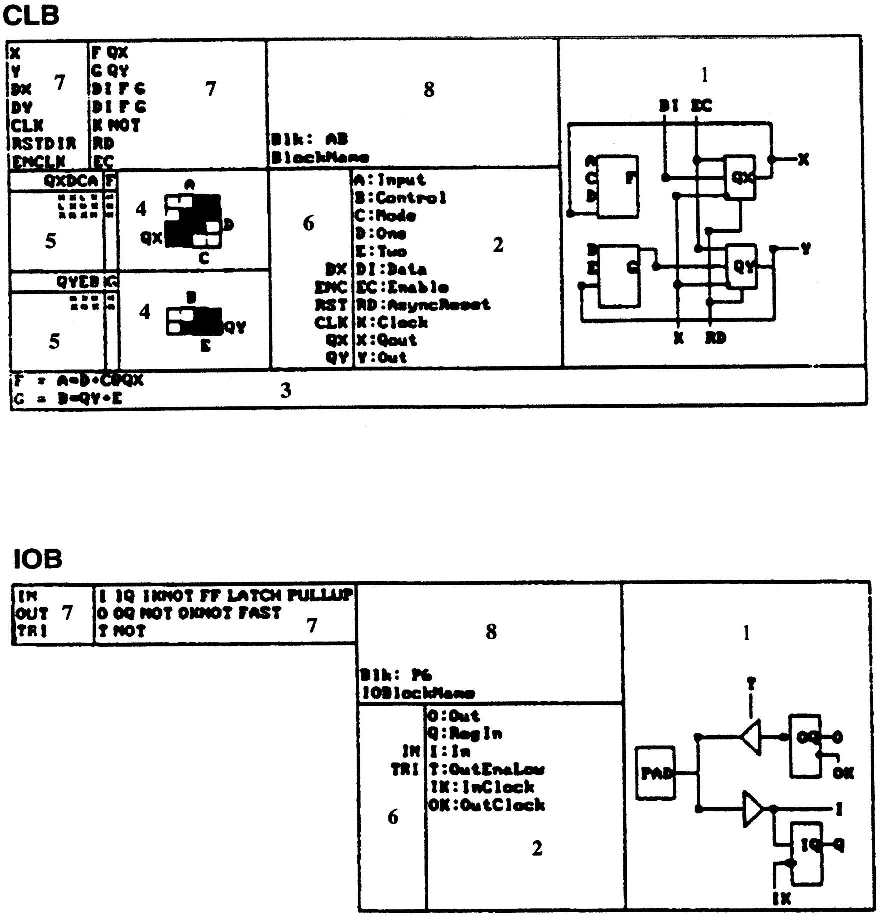


Figure 3-18 Block Editor Displays (XC2000)

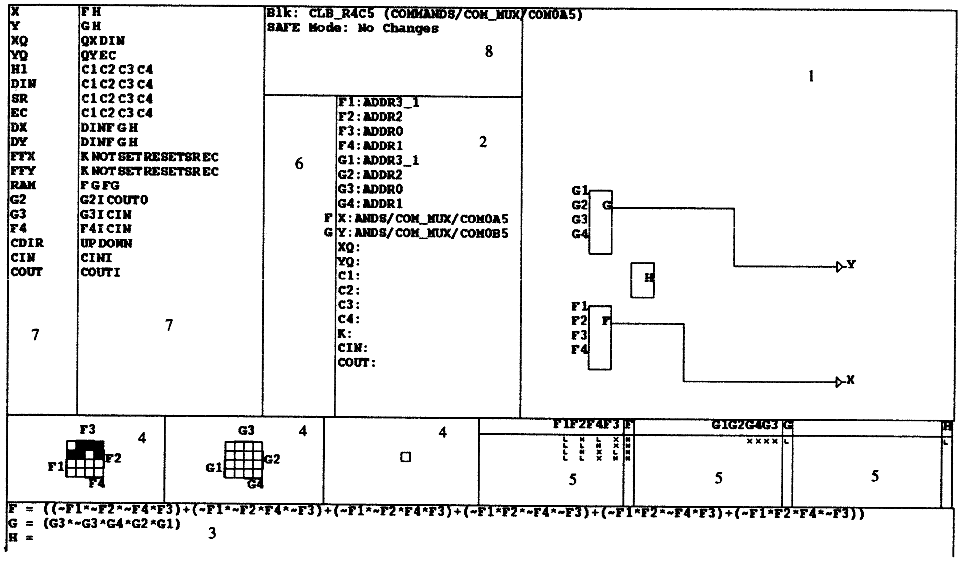


CLB

IOB

Figure 3-19 Block Editor Displays (XC3000)

CLB



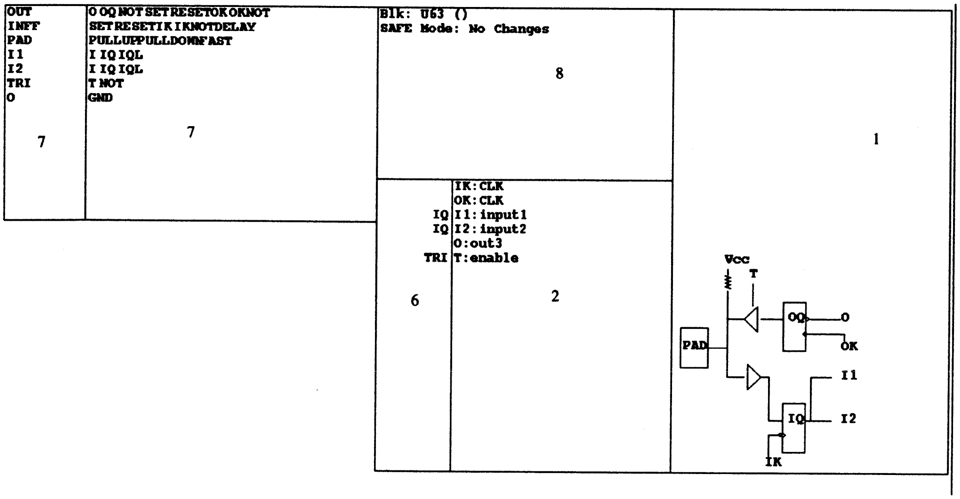
X	F H
Y	G H
XQ	QXDIN
YQ	QYEC
H1	C1 C2 C3 C4
DIN	C1 C2 C3 C4
SR	C1 C2 C3 C4
EC	C1 C2 C3 C4
DX	DINF G H
DY	DINF G H
FFX	K NOT SET RESET SR EC
FFY	K NOT SET RESET SR EC
RAM	F G FG
G2	G2I COUT0
G3	G3I CIN
F4	F4I CIN
CDIR	UP DOWN
CIN	CINI
COUT	COUTI

```
Blk: CLB_R4C5 (COMMANDS/COM_MUX/COM0A5)
SAFE Mode: No Changes
```

```
  F1:ADDR3_1
  F2:ADDR2
  F3:ADDR0
  F4:ADDR1
  G1:ADDR3_1
  G2:ADDR2
  G3:ADDR0
  G4:ADDR1
F X:ANDS/COM_MUX/COM0A5
G Y:ANDS/COM_MUX/COM0B5
  XQ:
  YQ:
  C1:
  C2:
  C3:
  C4:
  K:
  CIN:
  COUT:
```

```
F = ((~F1*~F2*~F4*F3)+(~F1*~F2*F4*~F3)+(~F1*~F2*F4*F3)+(~F1*F2*~F4*~F3)+(~F1*F2*~F4*F3)+(~F1*F2*F4*~F3))
G = (G3*~G3*G4*G2*G1)
H =
```

IOB



OUT	O OQ NOT SET RESET OK OKNOT
INFF	SET RESET IK IKNOT DELAY
PAD	PULLUP PULLDOWN FAST
I1	I IQ IQL
I2	I IQ IQL
TRI	T NOT
O	GND

```
Blk: U63 ()
SAFE Mode: No Changes
```

```
    IK:CLK
    OK:CLK
 IQ I1:input1
 IQ I2:input2
    O:out3
TRI T:enable
```

Figure 3-20 Block Editor Displays (XC4000)

Tags

The upper left corner of the Block Editor display shows the block tags. A tag is a block characteristic that can be configured one of several ways, such as the source of the X output or the function of the storage element. Current tag settings are highlighted.

Current tag settings are also shown in part by the net/tag association area of the display to the immediate left of the net assignment area.

In the XC2000 Block Editor, the symbolic figure to the right of the tag display illustrates the tag settings symbolically. For example, if the X tag is set to F, the figure shows the output of logic element F connected to the X output. For XC2000 CLBs, the base configuration (two functions of three variables, one function of four variables, or two functions joined by multiplexer) determines the basic blocks of this figure.

In the XC3000 Block Editor, the symbolic figure in the lower right of the display illustrates the tag settings symbolically. For example, if the X tag is set to F, the figure shows the output of logic-element F connected to the X output. For XC3000 CLBs, the base configuration (two functions of four variables, one function of five variables, or two functions joined by multiplexer) determines the basic blocks of this figure.

The figure center shows all of the block pins and the nets, if any, to which they belong. The area immediately to the left of the net and pin designation shows the internal signal that drives the X and Y outputs and the flip-flop/latch clock.

For CLBs, the area below the tag display shows the truth table and Karnaugh map for one or two functions depending on the base configuration of the CLB, if not a 5-variable XC3000 function. The equation that corresponds to the truth table and Karnaugh map appears along the bottom of the display. The equation, truth table, and Karnaugh map are updated each time a logic function is changed.

In the XC4000 Block Editor, the symbol figure to the far right of the display illustrates the tag settings symbolically. For example, if the X tag is set to H, the figure shows the output of logic element H connected to the X output. For XC4000 CLBs, the base configuration is always set to FG, as shown in Figure 3-20.

The center of the XC4000 Block Editor display shows all of the CLB elements and the nets, if any, to which they belong. The area immediately to the left of the net and pin designation shows the internal signal that drives the pin. In the above example, an H would appear to the left of X:netname.

For CLBs, three rectangular areas located in the lower left of the display are devoted to Karnaugh maps of the function generators F, G, and H. To the right of these maps is a truth table representation of F, G, and H. The equation that corresponds to the truth table and Karnaugh map appears along the bottom of the display. The equation, truth table, and Karnaugh map are updated each time a logic function is changed.

Command Entry Overview

You can enter commands and parameters in one of two ways.

- Use the cursor and mouse select button to select the displayed command or parameter.
- Type the command and/or parameter name and press ↵ or Enter key.

Mouse Selection of a Command

You are prompted for each parameter such as the name of a net. For many commands, you are prompted for additional parameters indefinitely. For example, the Addpin command continually prompts you for another pin name. To end the command and return to the `Cmd:` prompt, either select the Done option with the mouse or press the Enter or ↵ keys. You can also end the command if you enter more than one parameter at a time. The command line interpreter assumes the list is complete when you type in two or more pin names to the Addpin prompt.

Keyboard Entry of a Command

You can also type the parameters, separated by spaces, before pressing the ↵ or Enter keys. If you type all necessary parameters, you are no longer prompted for more parameters. For example:

```
addpin net1 ha.c
```

adds pin ha.c to net net1, and no other prompt appears.

Parameter Selection

You can select parameters with the mouse. When prompted for a parameter, such as a pin, block, net, or PIP, point to it with the cursor and verify that the desired selection appears in the cursor status line. When you are satisfied that the selection is correct, press the mouse Select button.

Default Values

Several commands include parameters that are given default values if you do not specify a parameter. For example, when you create a net using the AddNet command, XDE creates a default net name. If you only press the Enter or ↵ keys in response to the prompt for the net name, the XDE program uses the default net name. It also uses the default if you type the command without the optional parameter. For example:

```
addnet ha.c ha.d ha.k
```

adds pins HA.C, HA.D, and HA.K to a net with a default net name.

Pin Names

When specifying the pin names on the current block, you can use the single-letter name without the block-name prefix.

For XC20XX CLBs use A, B, C, D, K, X, or Y

For XC30XX CLBs use A, B, C, D, E, DI, EC, K, RD, X, or Y

For 40XX CLBs use F1, F2, F3, F4, G1, G2, G3, G4, C1, C2, C3, C4, K, X, XQ, Y, YQ

For XC20XX IOBs use T, I , K, or O

For XC30XX IOBs use O, T, OK, I, Q, or IK

For XC40XX IOBs use O, T, OK, IK, I1, I2

The current-blocks specified pin is assumed.

Block Names

When a current block is defined, you can enter a dot (.) instead of the current block name for all prompts requiring a block-name response.

Abbreviations

You can abbreviate most commands and parameters. See the "Command Summary" or "Command Descriptions" section later in this chapter.

Wildcards

A wildcard is a symbol that can represent any character or characters. The wildcard you use in XDE commands is the asterisk (*). If you enter G* as a block name, for example, the Design Editor performs the command on all blocks whose names begin with the letter G, including all blocks in row G, and any blocks that have been assigned names beginning with G.

You can use wildcards to specify parameters in the following commands.

- Alignpin
- Unroute pin
- Colornet
- Delblk
- FlagIOB
- Hilight
- Marknet
- Queryblk
- Route
- Routeblk
- Undefineprobe
- Unhilight
- Unrouteblk
- Unshownetconn
- Clearblk
- Colorblk
- Delnet
- Flagnet
- Markblk
- Markpin
- Querynet
- Routepin
- Shownetconn
- Showblkconn
- Unroute
- Unshowblkconn
- Weightnet
- XDelay

The commands Clearblk, Unroute pin, Delblk, Delnet, Routeblk, Marknet, Markpin, Markblock, FlagIOB, Weightnet and Unroute display the prompt, `ARE YOU SURE?(Y/N)` when you use a wildcard. This prompt safeguards against accidental deletions of data.

You can use any number of wildcards to specify parameters. For example, the expression *A*B* can represent Marble, Parabola, Halfback. A single asterisk represents all instances of a parameter. For example, Route * routes all nets.

Editor Commands

NOTE *With the introduction of XDE 4.1, a Safe mode is available. In this mode, you cannot make logic changes to your design in the XDE Design Editor. To make such changes, set the mode to Expert.*

This section lists the commands alphabetically and describes their use. Several commands are available from more than one program but appear on the same menu name. For example, the KeyDef command appears on the Profile menu in the MakeBits program and the EditLCA program. The command syntax is given in the following format:

- Command parameters are listed after the command name. Table 3-10 lists the abbreviations and the type of parameter needed.

Table 3-10 Vaild Command Parameter Abbreviations

Abbreviation	Parameter
name	new net name or block name
net	existing net
pin	pin on a block blk
blk	block CLB or block IOB
PIP	switch point or switching matrix pin
option	any specific option

- Optional parameters are shown in italics.
- An Ellipsis (...) following a parameters indicate that you can enter any number of that parameter.
- Any abbreviations are listed below the command syntax.

Several commands prompt for additional parameters indefinitely. You can end these commands by pressing the ↵ or Enter key, selecting Done, or by pressing a mouse button defined as Done.

Several commands result in the configuration or reconfiguration of interconnect if the Autoroute option is on. This effect is noted where applicable.

Several commands result in the deconfiguration of spurs. A spur is a configured interconnect that, because of deletions, no longer connects two pins or active PIPs. This deconfiguration is noted where applicable.

Block Editor commands are performed only on the current block. These commands are indicated by the word Block after the command name.

Addnet — Create a Net

Menu	Pin
Syntax	addnet *netname pin ...*
Abbreviations	addn, an
See also	Addpin, Editnet, Delnet, Namenet, Route, Autoroute

If you do not specify a net name, XDE uses a default net name, consisting of net followed by a number. If you delete a net that has a default net name, its name is not reused.

You can only enter pins that do not already belong to a net. If you select Cancel, the net is not created. If you enter no pins, an empty net is created. If the Autoroute option is on, the net is routed when you end the command. You must enter either the net name or pins.
For example.

```
addnet ad.c ae.x
```

creates a net with a default net name that contains pins AD.C and AE.X.

Figure 3-21 illustrates the results of entering the following command syntax:

```
addnet demo ca.a ba.x cb.b cc.c
```

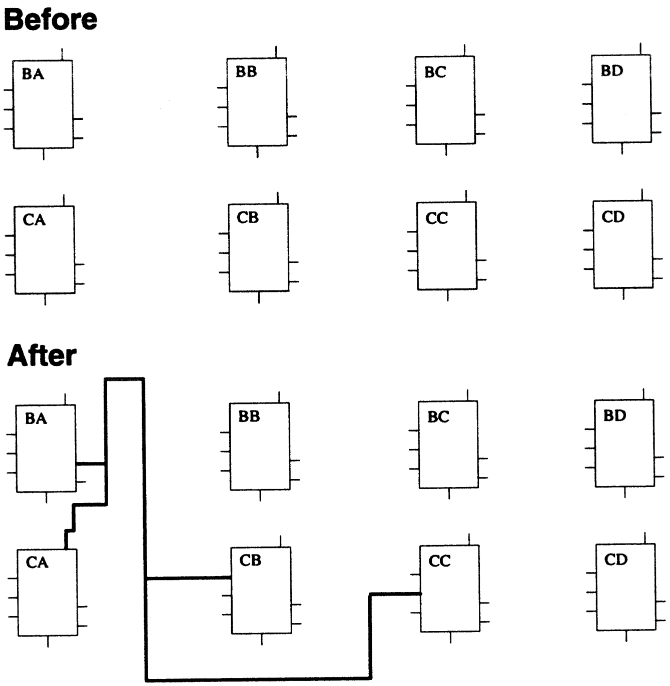


Figure 3-21 Addnet Example (with Autoroute on)

Addpin — Add a Pin to a Net

Menu	Pin
Syntax	addpin *net pin ...*
Abbreviations	addp, ap
See also	Addnet, Delpin, Unroute pin, Routepin

This command adds specified pins to the specified net. If the net does not exist, it is created, and then this command functions the same as Addnet. You can only enter pins that do not already belong to a net. If the Autoroute option is on, the pins are routed after the command is ended. You can cancel the Addpin command at any time by selecting the Cancel option with the mouse.

Alignsig — Assign a Net to a Chosen Pin Within Blocks

Menu	Pin
Syntax	alignsig *net pin block...*
Abbreviations	al
See also	Swapsig

This command is a shorthand for multiple Swapsig commands. Alignsig is most useful when you pre-place blocks to align pins to take advantage of the Longline resources. For each specified block, the Alignsig command changes the functionality of the specified pin with the pin that contains the chosen net.

Assignprobe — Assign an Internal Net to a Predefined Probe

Menu	Probe
Syntax	assignprobe *probe net*
Abbreviations	assignpr
See also	Defineprobe, Unassignprobe, Queryprobe, Readprobe, Saveprobe, Debugload, Download, MakeBits, Port

Use this command to assign an internal net to a predefined probe (test point) associated with a particular IOB. XDE automatically tries to route the net associated with the probe. It then reports the delay from the net source to the probe, including IOB delay.

In a typical circuit to be debugged, you should leave the oscilloscope probes connected to the same pins. By assigning a net to a probe, you can look at the internal signal through the probe. If a different net is to be analyzed, you need only reassign the probe to a new net.

First, Assignprobe prompts for a probe name. It presents a list of valid probes that have been defined. Select a probe either by clicking in the menu, or by typing the name of the probe. Next, you are prompted for a net to connect to the probe. A menu displays a list of nets only if you have Netmenus on. You can select a net either by clicking in the menu, pointing within the graphic editor screen at the net and clicking on it, or typing the name of the net. After this procedure, the probe is assigned to that particular net.

NOTE *After multiple nets are routed to multiple probes, there can be some differential in the delay from each net to its probe which can change the time at which signals are seen in reference to one another. On an oscilloscope, the XDE router tries to reduce these delays to a minimum, but the skew is also important. See Queryprobe for information regarding the relative timing differences of probes.*

WARNING!!! *Using a probe changes the internal net delays. The XDE router tries to minimize the change in delay, but the addition of a probe almost always slows down the observed net. Be aware of this when trying to analyze high-speed designs, where either delay skew or total delay is critical.*

Autodrc — Turn Automatic DRC Checking On/Off During Editnet

Menu	Profile
Syntax	autodrc on|off
Abbreviations	autod

This command enables or disables automatic design rules checks during Editnet. When you route a net with Editnet, XDE checks the correctness of each added PIP if the Autodrc option is on.

Autoroute — Turn Automatic Routing On or Off

Menu	Profile
Syntax	autoroute on|off
Abbreviations	autor
See also	Route

When this option is on, the corresponding interconnect is automatically configured when you add pins to a net. If the option is off, you can route nets using the Route command. The valid options are on and off. The initial value for this command is determined by the editlca.pro file.

Autotime — Turn Automatic Timing On or Off

Menu	Profile
Syntax	autotime on|off
Abbreviations	autot
See also	Delay, Querynet

Autotime on enables automatic timing. Autotime off disables automatic timing. With Autotime on, all commands that alter the configuration of nets cause the delays for the load pins on those nets to be automatically recalculated at the end of the command. The delays for the load pins are displayed on the cursor-status line and in the Querynet command.

Base — Change the Base Configuration

Menu	Base
Syntax	base *option*
Abbreviations	none

The Base command deconfigures the current block and sets the base configuration of the block. Table 3-11 lists the valid options for the base command.

Table 3-11 Base Command Options

Basename	Usage
IO	For IOBS Only
F	One function of up to four variables for XC2000LCA devices, five variables for XC3000 LCA devices.
FG	Two functions of up to three variables each for XC2000 LCAs, four variables each for XC3000 LCAs, or two independent functions of four variables each for XC4000 LCA devices
FGM	Same as FG, but the two function outputs are multiplexed together and controlled by the B input for the XC2000 series, or the E input for the XC3000 series.

Base FG is the only base used for XC4000 CLBs. Figure 3-22 illustrates four base configurations: three for a CLB and one for an IOB. These symbolic figures are those that appear for unconfigured blocks. As you configure the block, more symbols are added to reflect the configuration.

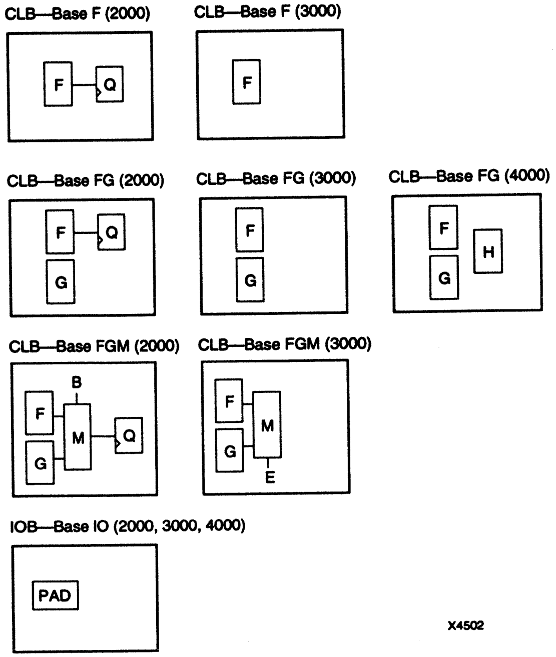


Figure 3-22 Base Configurations

Blkcolors — Assign Colors to Block Editor Display

Menu	Screen
Syntax	blkcolors *item color*
Abbreviations	blkc, bc
See also	Piecolors

This command is only available in 8 or 16 color modes. It assigns colors to most block-editor display features. You can change the following display elements.

Table 3-12 Valid Blkcolors Command Elements

Item	Default Color	Description
Reset	N/A	Restores default settings
Kmap0color	BLACK	Karnaugh Map 0 (off)
Kmap1color	YELLOW	Karnaugh Map 1 (on)
Kmaplabel	GREEN	Karnaugh Map label
Kmapborder	GREEN	Karnaugh Map border
Bordercolor	GREEN	Region border
Equationcolor	RED	Equation
SDblockcolor	GREEN	Schematic diagram block
SDnetcolor	YELLOW	Schematic diagram net
SDpincolor	GREEN	Schematic diagram pin
SDlabelcolor	YELLOW	Schematic diagram label
Tagcolor	YELLOW	Configuration table (Left) tag
Ontagcolor	YELLOW	Configuration table (Right) “on” tag
Offtagcolor	GREEN	Configuration table (Right) “off” tag
NBpincolor	YELLOW	Net Binding table pin
NBnetcolor	GREEN	Net Binding table net
NBtagcolor	YELLOW	Net Binding table tag
TTlabel	YELLOW	Truth Table label
TTcube	GREEN	Truth Table cube
TTvalue	YELLOW	Truth Table value

If XDE does not find a color for an item in the profile (xact.pro file) while loading a design, it uses the default color.

BlkMenus — Display Scrolling Menus of Blocks

Menu	Screen Profile
Syntax	blkmenus on|off
Abbreviations	none
See also	Netmenus

When the BlkMenus option is enabled, a scrolling menu of block names appears when you execute commands that require block selection.

Cdata — Display Configuration Information for a Block

Menu	Config
Syntax	cdata
Abbreviations	cd
See also	Queryblk

This command displays the current block configuration in text form and the Editor commands that would result in this configuration. The output includes a Base command, a Config command, and possibly one or more Equate commands.

Examples of what might appear in the Config command follow.

The following syntax indicates that logic function F is configured with inputs A and D.

```
F:A:D
```

The following syntax means that the storage element is configured as a latch (XC2000 only).

```
Q:LATCH
```

The following syntax is an empty tag; the X output is not configured.

```
X:
```

Clear — Deconfigure a Tag Setting or Equation

Menu	Config
Syntax	clear *tag*
Abbreviations	cl
See also	Clearblk, Config

This command clears the equations and tags associated with CLBs and IOBs. When you enter the command, a list of choices applicable to the type of block appears. The tag you enter is cleared.

To clear all tag settings, use the Clearblk command.

Clearblk — Deconfigure a Block

Menu	Blk
Syntax	clearblk *block*
Abbreviations	clearb, clb
See also	Delblk, Editblk

This command returns the specified block to its unconfigured state. Clearblk clears any tags that were previously set within the block. Clearblk initializes the base to FG, but the block name is retained.

Clearmargins — Clear Margin Delays Created by Setmargin

Menu	Timing
Syntax	clearmargins
Abbreviations	clearm, clm
See also	Setmargins, Querymargins, Savemargins

Clearmargins clears all the margin delay for each path set by the Setmargins command and added to the calculated on-chip delays. Clearmargins clears all of the margin delays that were previously set with the Setmargins command.

ClearTemplate — Clear Defined XDelay Options

Menu	Timing
Syntax	cleartemplate
Abbreviations	clt
See also	QueryTemplate, SaveTemplate, ReadTemplate

ClearTemplate clears any previously defined XDelay options and restores them to their default.

Cmtblk — Tag a Block with Comment Text

Menu	Blk
Syntax	cmtblk *block comment*
Abbreviations	cmtb
See also	Editblkcmt

This command tags the specified block with the given text. If no text is given, the comment associated with the specified block is removed. The Block Editor displays as much of the comment text as possible. Comment text entered in this manner is preserved in the LCA file.

Colorblk — Assign a Color to a Block

Menu	Blk
Syntax	colorblk *color block*
Abbreviations	colorb

This command is only available in 8- or 16-color modes. If the XDE program finds a block assigned a color not currently available in the palette, while loading a design, it gives that block the default color. Blocks with unavailable colors can result from assigning block colors while using a 16-color palette and then later reading in the design while using a palette with fewer colors available.

Colornet — Assign a Color to a Net

Menu	Net
Syntax	colornet *color net*
Abbreviations	colorn

This command is only available in 8- or 16-color modes. If the selected net is already routed, the XDE program redraws it in the chosen color. If the selected net is unrouted, XDE records the new color but waits to assign it until that net is routed. If the XDE program finds a net assigned with a color not currently available in the palette while loading a design, it gives that net the default color. Nets with unavailable colors can result from assigning net colors while using a 16-color palette and then later reading in the design while using a palette with fewer colors available.

Config — Configure a Block Logic and Connections

Menu	Config
Syntax	config *tag:setting ...*
Abbreviations	con
See also	Clear, Equate, Order

Tags are block characteristics that can be set to one of several options. These options are listed in Figure 3-23 through Figure 3-25 and explained in the text that follows. CLB-tag options depend on the base configuration (F, FG, or FGM) for XC2000 and XC3000 designs.

To assign tag settings, enter the Config command and type the tag name, a colon, and the option name for each tag. No spaces are permitted between the tag and the colon, or the colon and the setting. For example, to connect the Set input of an XC2000 series storage element to the A input, type `Config Set:A`. The assigned option changes color. If you use the mouse, you do not have to enter the Config command; just point to the displayed option with the cursor and press the mouse Select button.

There are three ways to clear a tag setting.

- Type Config, the tag name, and a colon, and press ↵ without entering a tag setting.
- Place the cursor on the option and press the select button.
- Use the Clear command.

Tag	Options
CLB	
X	F, G, M or Q
Y	F, G, M or Q
Q	FF or LATCH
SET	A, F or M
RES (RESET)	D, F, G or M
CLK	RD
F	EC
G	A, B, C, D, or Q
For Base F	Function G = F
For Base FGM	Only M, not Functions F and G, are individually available
IOB	
I	PAD or Q
BUF (BUFFER)	ON or TRI

Figure 3-23 XC2000 Tag Options

Tag	Options
CLB:	
X	F or QX
Y	G or QY
DX	DI, F or G
DY	DI, F or G
CLK	K and NOT
RSTDIR	RD
ENCLK	EC
For Base F	Function G = F
For Base FGM	Only M, not Functions F and G, are individually available
IOB:	
IN	I, IQ, IKNOT, FF/LATCH, PULLUP *
OUT	O, OQ, NOT, OKNOT, FAST
TRI	T and NOT

Note:

* The In = Pullup option is not permitted when the IOB is configured as an output.

Figure 3-24 XC3000 Tag Options

Tag	Options
CLB	
X	F or H
Y	G or H
XQ	QX or DIN
YQ	QY or EC
H1	C1, C2, C3, or C4
DIN	C1, C2, C3, or C4
SR	C1, C2, C3, or C4
EC	C1, C2, C3, or C4
DX	DIN, F, G or H
DY	DIN, F, G or H
FFX	K, NOT, SET, RESET, SR and EC
FFY	K, NOT, SET, RESET, SR and EC
RAM	F, G or FG *(CLB memory functions)*
G2	G2I or COUT0 *(Carry Logic)*
G3	G3I or CIN *(Carry Logic)*
F4	F4I or CIN *(Carry Logic)*
CDIR	UP or DOWN *(Carry Logic)*
CIN	CINI *(Carry Logic)*
COUT	COUTI *(Carry Logic)*
IOB	
OUT	O, OQ, NOT, SET, RESET, OK and OKN
INFF	SET, RESET, IK, IKNOT and DELAY
PAD	PULLUP, PULLDOWN and FAST
I1	I, IQ, or IQL
I2	I, IQ or IQL
TRI	T and NOT
O	GND *(IOB internal ground)*
OSPEED	FAST, MEDFAST, MEDSLOW, SLOW

Figure 3-25 XC4000 Tag Options

NOTE

OSPEED only applies to XC4000A devices.

Most CLB tags determine the source of the tag signals. The other tags determine special configuration information. Most options are mutually exclusive. For example, if the X tag of an XC3000 CLB is set to QX, you can change its setting to F by typing Config X:F, or by selecting the F option with the mouse. Either method automatically deselects the QX option. The Not option for the CLK tag acts as an inverter (FFs are clocked on the falling edge) and is not mutually exclusive of the other CLK options.

The F and G tags do not appear in the tag display. The base configuration of the CLB determines their connectivity and functionality. The Karnaugh maps represent the function F and G will perform. A filled-in cell represents a one (1), and an empty cell represents a zero (0). You can add ones to or remove ones from the Karnaugh map by pointing at a particular cell in the map and clicking the mouse Select button. When you alter the Karnaugh map, the Boolean equation is updated at the bottom of the Block Editor screen. The equations can also be specified by using the Equate command or by typing *function* = Boolean expression, where *function* is equal to F, G, or for an XC4000 CLB, H.

CLB Tag Options (XC2000 Devices)

For an XC2000 CLB, the Q-tag setting determines whether the CLB storage element acts as a latch or a flip-flop. You can combine the Not option with K, C, F, M, or G. For example, to configure the clock input to be the inverse of G, type Config CLK:G:NOT.

If the base configuration is FG, you can configure F and G as functions of up to three variables. If the base configuration is F, F can be a function of up to four variables, with G = F. Base FGM produces M = F if B is 1 (High), and M = G if B is 0 (Low).

CLB Tag Options (XC3000 Devices)

Storage elements for XC3000 CLBs are always flip-flops. The RSTDIR and ENCLK tags activate the direct (asynchronous) reset and clock enable, respectively. When RSTDIR is High, the output of the flip-flop is set to zero. When ENCLK is not selected, or is selected and driven High, the flip-flop clock is active and can register data. When the ENCLK tag is selected and the net driving it is low, the present state of both flip-flops in the CLB is recirculated regardless of clock transitions.

If the base configuration is FG, F and G can be configured as functions of up to four variables. If the base configuration is F, F can be a function of up to five variables. Base FGM produces M = F if E is 0 (Low), and M = G if E is 1 (High).

CLB Tag Options (XC4000 Devices)

Storage elements for XC4000 CLBs are always flip-flops. The EC option of the CLK tag works similarly to the ENCLK option of the XC3000 CLB. If the SR option is on, the SR signal can drive the Set or Reset signal of each flip-flop. The Set and Reset options are mutually exclusive for each flip-flop, although you could drive the Set signal of one flip-flop and the Reset signal of the other with the SR signal. Set and Reset perform an asynchronous set or clear of the CLB flip-flop.

You can configure the RAM tag as F, G, or FG. Essentially, each half of an XC4000 CLB can be configured as a 16 x 1 RAM. For example, if you type Config RAM:F, the F function generator is configured as a 16 x 1 RAM, with the four address bits sourced from CLB inputs F1, F2, F3, and F4. The input data is sourced from DIN, and the output of F would be the data currently being addressed. The G function generator would be available for other uses. If the RAM tag is configured as G, the four address bits are sourced from G1, G2, G3, and G4. The input data is sourced by H1. If both F and G are turned on, two independent 16 x 1 RAMs are created, although they both have the same WE (write enable) signal. If the RAM tag is configured as FG, both F and G are used to create a 32 x 1 RAM, and the fifth address bit is sourced by H1. The input data is driven by DIN.

The tags G2, G3, F4, CDIR, CIN, and COUT are special flags used to determine the configuration of the XC4000 CLB carry logic (Figure 3-25). The tag G2 selects the normal CLB input (G1I) or COUT0 from the carry logic. Similarly the G3 tag selects between G3I and CIN, and F4 selects between F4I and CIN. CDIR specifies the carry direction, that is, which way the carry logic is to propagate through the CLB. The CIN and COUT tags determine which signal drives the CarryIn and CarryOut of the CLB. If CINI or COUTI is selected, the corresponding input to the function generator is driven by the carry logic. In some carry logic configurations, these inputs may also be driven by external signals, depending on the configuration. (See the following ConfigCarry section for a list of carry logic configurations.)

NOTE

The Config command does not change the equation for F or G if the setting matches the variables already in the equation, for example.

```
F = A + B
Config F:A:B     (Does not alter equation)
Config F:A:B:C   (Does alter equation)
```

Also, the Config command can be used to order variables. Config F:A:C:B is the same as Order F A C B.

IOB Tag Options (XC2000 Devices)

The XC2000 I-tag setting determines whether the IOB routes input directly from the pad or through the input latch. The BUF tag sets the state of the output buffer. If it is on, the output buffer is always activated, and the IOB acts as an output. If it is set to TRI, the state of the IOB 3-state (T) input turns the output buffer on and off, enabling bidirectional signals.

The XC3000 I and IQ tags enable either or both the direct and registered input from the package pin. The IK provides positive-edge flip-flop clocking and active-Low transparent latch. IKNOT inverts the active sense of the input storage elements. All IK senses (positive-edge or negative-edge) on a die edge must be the same. The flip-flop/latch determines whether the storage element is an edge-clocked or level-transparent device. Pull-up enables the pin pull-up resistor, which defines input levels of unused package pins. Pull-up is not allowed when the output buffer is used.

IOB Tag Options (XC3000 Devices)

The XC3000 O and OQ tags select either a nonregistered or registered output that the IKNOT might invert. The OKNOT selects the inverted sense of the output flip-flop clock, and as with the IK, all OK senses on a die edge must match. The Fast tag adjusts the output-pin slew rate for optimal speed or provides a controlled slew rate for reduced system noise. 3-state output and the logic sense of the control signal are determined by TRI:T and TRI:T:NOT. TRI:T:NOT selects the active-Low to provide a high-impedance output buffer. The normal sense is a High level and implies a high impedance (active-Low output enable).

Clocking Restrictions

The XC3000 IOBs along the same edge of the die cannot be independently clocked. You must ensure that IOB clock pins along the same edge of the die and on the same net meet the following criteria.

- All flip-flop clock polarities must be the same.
- All latch-clock polarities must be the same.
- The latch-clock polarity must be opposite of the flip-flop clock polarity if both flip-flops and latches are used along the same edge.

IK*	Reset**	IQ***
Flip-flop		
X	0	Pad Data
Don't care	1	0
Latch		
1	0	Transparent
Don't care	1	0

* EnableClock recirculates flip-flop states only if the tag is selected and the control net is Low.

** CLB Reset affects the CLB flip-flops only if the tag is selected and the control net is High. All CLB flip-flops are reset by the configuration process and by the external Reset pin.

*** IOB storage elements are cleared by the configuration process and by the external Reset.

Figure 3-26 XC3000 IOB Input Storage Element (Non-inverted Clock IK)

IOB Tag Options (XC4000 Devices)

The XC4000 Out tag determines how the IOB output is to be configured. You can set the Out tag to the O option to enable the direct output of the IOB. For both Out and InFF, the clock (OK) to the output flip-flop (OQ) and the clock (IK) to the input flip-flop/latch (IQ,IQL) can be inverted by selecting OKNOT and/or IKNOT, respectively. You can use the Set or Reset options to program the flip-flop to power up in a known initial state. The Delay option on InFF allows the input signal to be delayed, which can eliminate potential hold-time problems on the input flip-flop.

You can configure the inputs I1 and I2 as the direct input I, the registered input IQ, or the latched input IQL. The PAD tag can be configured as Pullup, Pulldown, or Fast. The Pullup and the output buffer can both be enabled, although the Pullup resistor has a high impedance. The Pulldown option has been added to provide a low signal at the pad when the output driver is disabled.

The 3-state control (TRI) to the output driver is enabled with the T option, which is an active-Low enable signal. When both T and Not are on, the 3-state signal is an active-High enable.

The O tag, when enabled, serves to ground the input to the output buffer. This can be used if a logic 0 is needed externally, or if an open-collector type output is desired. In this case, the 3-state signal would be driven by the open-collector input, and either a logic 0 or high impedance would be provided at the output.

Configcarry — Configure a CLB Carry Logic Function (XC4000 Only)

Menu	Config
Syntax	configcarry *option*
Abbreviations	Hcc, configc

This command configures the specified CLB (XC4000 family only) so that the carry logic performs a certain function. If the Unused option is chosen, the carry logic is not used within the CLB.

The valid options are as follows. For more details about each option, refer to the *XACT Libraries Guide*.

```
add-f-ci       force-0     dec-g-f3-  sub-g-f1
add-fg-ci      force-1     inc-f-ci   sub-g-ci
add-g-f1       force-f1    inc-fg-ci  sub-g-f3-
add-g-ci       force-ci    inc-fg-1   incdec-f-ci
add-g-f3-      force-f3-   inc-g-1    incdec-fg-ci
addsub-f-ci    dec-f-ci    inc-g-f1   incdec-fg-i
addsub-fg-ci   dec-fg-ci   inc-g-ci   incdec-g-0
dec-fg-0       inc-g-f3-   incdec-g-f1
addsub-g-f1    dec-g-0     sub-f-ci   incdec-g-ci
addsub-g-ci    dec-g-f1    sub-fg-ci
addsub-g-f3-   dec-g-ci    sub-g-1    unused
```

The option name most often contains three fields separated by dashes.

The first field is the function type.

add	Adder	**inc**	Incrementer
sub	Subtracter	**dec**	Decrementer
addsub	Adder/Subtracter	**incdec**	Incrementer/Decrementer

Force sets the Carryout to some value.

The second field normally indicates which function generators in the CLB are being used in the configuration. Table 3-13 lists the valid options.

Table 3-13 Configcarry Command Options

F	F function generator and/or X register used. Typically the MSB of a macro function.
G	G function generator and/or Y register used. Typically the LSB of a macro function.
FG	Both function generators and/or registers used. Typically the middle of a macro function.

The third field (or the second, in the case of Force) indicates how the CarryIn signal is to be connected. The valid options are as follows.

CI	Carry in is sourced from adjacent CLBs.
F1	Carry in is sourced by F1 input pin.
F3	Carry in is sourced by inversion of F3 input pin.
0	Carry in is tied permanently Low.
1	Carry in is tied permanently High.

Configure — Assign MakeBits Configuration Options

Menu	Probe
Syntax	configure *option*
Abbreviations	config

In the MakeBits program, configure options are normally set with the same Configure command, or by pointing and clicking with the mouse.

In XDE, you can use Configure to change these options without exiting the Editor. These options vary with the family (XC2000, XC3000, or XC4000). For details, refer to the "The MakeBits Program" chapter in Volume 2 of the *XACT Reference Guide*.

Copyblk — Copy the Configuration of One Block to Another

Menu	Blk
Syntax	copyblk *sourceblk destblk*
Abbreviations	copyb, cb
See also	Moveblk, Swapblk

This command copies the internal configuration of Sourceblk to each of the Destblk blocks. Sourceblk must be a configured block, and each Destblk must be unconfigured. Net connections are not affected. All blocks must be of the same type, either all CLBs or all IOBs.

Cursor — Change the Cursor Shape

Menu	Profile
Syntax	cursor *option*
Abbreviations	cu
See also	Mouse

There are four cursor shape options.

Arrow	A slanted arrow whose tip is the cursor location
Bug	a small X whose center is the cursor location
Cross	A large cross whose center is the cursor location
Gunsight	A screen-width cross centered at the cursor location

You can abbreviate these options to A, B, C, and G, respectively. The initial value of this command is set in the editlca.pro file.

Cut — Save Part of a Configuration in a File for Moving

Menu	Misc
Syntax	cut *filename block*
Abbreviations	none
See also	Macro, Paste, Cutmacro

Use the Cut command in conjunction with the Paste command to move a configuration from one area of an LCA to another. To do this, first store the configuration in a file using the Cut command. Next, delete the configuration of the original set of blocks using the Delblk command. Finally, configure the new set of blocks using the Paste command with the newly created file.

NOTE *Cut only works with designs from the same family. You cannot use Cut between an XC3000 design and an XC4000 design.*

The Cut command checks each block selected for cutting to see if it has already been selected. If the block has already been selected, its subsequent selection is rejected and an error message is displayed.

The Cut command creates a macro containing the following.

- The Design Editor commands necessary to reproduce the internal logic of the blocks you specify
- The Addpin commands that produce net connections identical to the net connections of the blocks you specify

- The Nameblk commands to rename blocks with the user-defined names of the blocks you specify, if these names exist
- The Parameter commands and corresponding % expressions you can enter for the blocks that will receive the macro

These commands are organized and stored in the specified file. The file must have the .cut extension. The file name is automatically given the .cut extension, if you do not enter it. If the CUT file name already exists, the Cut command renames it with the .ocf extension, displays a message that the file has been renamed, and then creates the new CUT file as requested.

You can use the CUT file to paste only a single instance of a configuration. You must delete the original set of blocks before configuring the new set, otherwise two events occur.

- Nameblk commands in the file fail because the block name is still assigned to the original block.
- Addpin commands in the file result in multiple source pins for each net, since each net has an existing source pin in the original set of blocks.

Cutmacro — Save a Macro Configuration for Other Locations

Menu	Misc
Syntax	cutmacro *file name block*
Abbreviations	cutm
See also	Macro, Cut, Paste

The Cutmacro command goes a step beyond the Cut command by creating a macro that you can use for multiple instances of the same configuration in one design.

When you select a block for cutting, XDE checks to determine if the specified block has already been selected. If the block has already been selected, its subsequent selection is rejected and XDE issues an error message.

All macros created by the Cutmacro command begin with these parameter commands.

- A parameter command specifies a name that identifies the instance of the configuration (an instance name). Enter a unique instance name each time you use the macro. Parameter commands specify the blocks that will receive the macro.

- Parameter commands specify the names of all nets that are inputs to the macro connected to a block in the macro but having source pins outside of the macro.

When you start the macro, the default names for these nets are the same as the names of the original nets. If you use these default names, and they do not already exist in the design, new nets are created for each instance of the macro. If you specify existing nets instead, the macro adds pins to the existing nets.

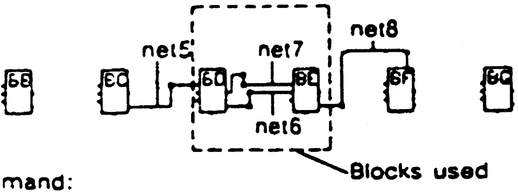
Macro outputs are named by appending the instance name to the beginning of their original name. This naming ensures that the net names of each instance are unique.

The Cutmacro command determines the following commands, using the blocks you specify.

- The Design Editor commands necessary to reproduce the internal logic of the blocks you specify.
- The Addpin commands that produce net connections identical to the net connections of the blocks you specify. The nets are named as described above.
- The Nameblk commands to rename blocks with the user-defined names of the blocks you specify, if these names exist. The instance name is appended to the end of all names used in Nameblk commands so that each instance results in uniquely named blocks.

These commands are organized and stored in the macro file whose name you specify. The file must have the .mac extension; it is automatically given the .mac extension if you do not enter an extension. If the MAC file name already exists, the Cutmacro command renames it with the .omf extension, displays a message that the file has been renamed, and then creates the new MAC file as requested.

An example configuration and the text of macros created using Cut and Cutmacro are shown in Figure 3-27.



Cut command:

```
Parameter CLB BD Select BD block:
Parameter CLB BE Select BE block:
Editblk %1
Base FG
Config X:Q Y:G F:A:B:C G:A:D Q:LATCH SET: RES: CLK:K
Equate F = A+B*C
Equate G = D@A
Endblk
Editblk %2
Base F
Config X:F Y: F:A:B:C:D Q: SET: RES: CLK:
Equate F = A+B+C+D
Endblk
Addpin net5 %1.B
Addpin net7 %1.X %2.B
Addpin net6 %1.Y %2.C
Addpin net8 %2.Y
```

CUTMACRO command:

```
Parameter NAME ? Enter instance name:
Parameter NET net5 Select net5 net:
Parameter CLB ? Select BD block:
Parameter CLB ? Select BE block:
Editblk %3
Base FG
Config X:Q Y:G F:A:B:C G:A:D Q:LATCH SET: RES: CLK:K
Equate F = A+B*C
Equate G = D@A
Endblk
Editblk %4
Base F
Config X:F Y: F:A:B:C:D Q: SET: RES: CLK:
Equate F = A+B+C+D
Endblk
Addpin %2 %3.B
Addpin %1net7 %3.X %4.B
Addpin %1net6 %3.Y %4.C
Addpin %1net8 %4.Y
```

Figure 3-27 Cut and Cutmacro Example

Debugload — Create a Bitstream and Download to Download Cable

Menu	Probe
Syntax	debugload
Abbreviations	debugl, dl
See also	MakeBits, Download, Port

This command performs, in sequence, the MakeBits and Download commands. Once the bitstream is created, Debugload automatically attempts to download it through the download cable. Use this command in conjunction with the Probe menu commands when you are debugging a circuit.

NOTE *Debugload works with the Download command, which does not work for the XChecker cable.*

Defineprobe — Assign Name to an IOB to Use as Test Point

Menu	Probe
Syntax	defineprobe *probe_name unused IOB*
Abbreviations	definepr
See also	Undefineprobe, Saveprobe, Assignprobe, Readprobe

This command allows a logical name to be associated with a particular IOB. The XDE Design Editor stores this association in memory. To save the currently defined probes, use the Saveprobe command.

Defineprobe prompts for a probe name, then displays a list of valid (unused) bonded IOBs. The list might have an Up arrow and/or a Down arrow as a selection. Select either of these to scan through the pages of the list. You can also select IOB by clicking on the main graphic screen or by typing it in.

For example, you can name a probe Channel_2 (corresponding to an oscilloscope probe), and associate this name with IOB P35. The oscilloscope probe can then be connected externally to P35, without the problem of having to move it from pin to pin. If you want to examine a different signal, just reassign the probe to a different internal net (see Assignprobe).

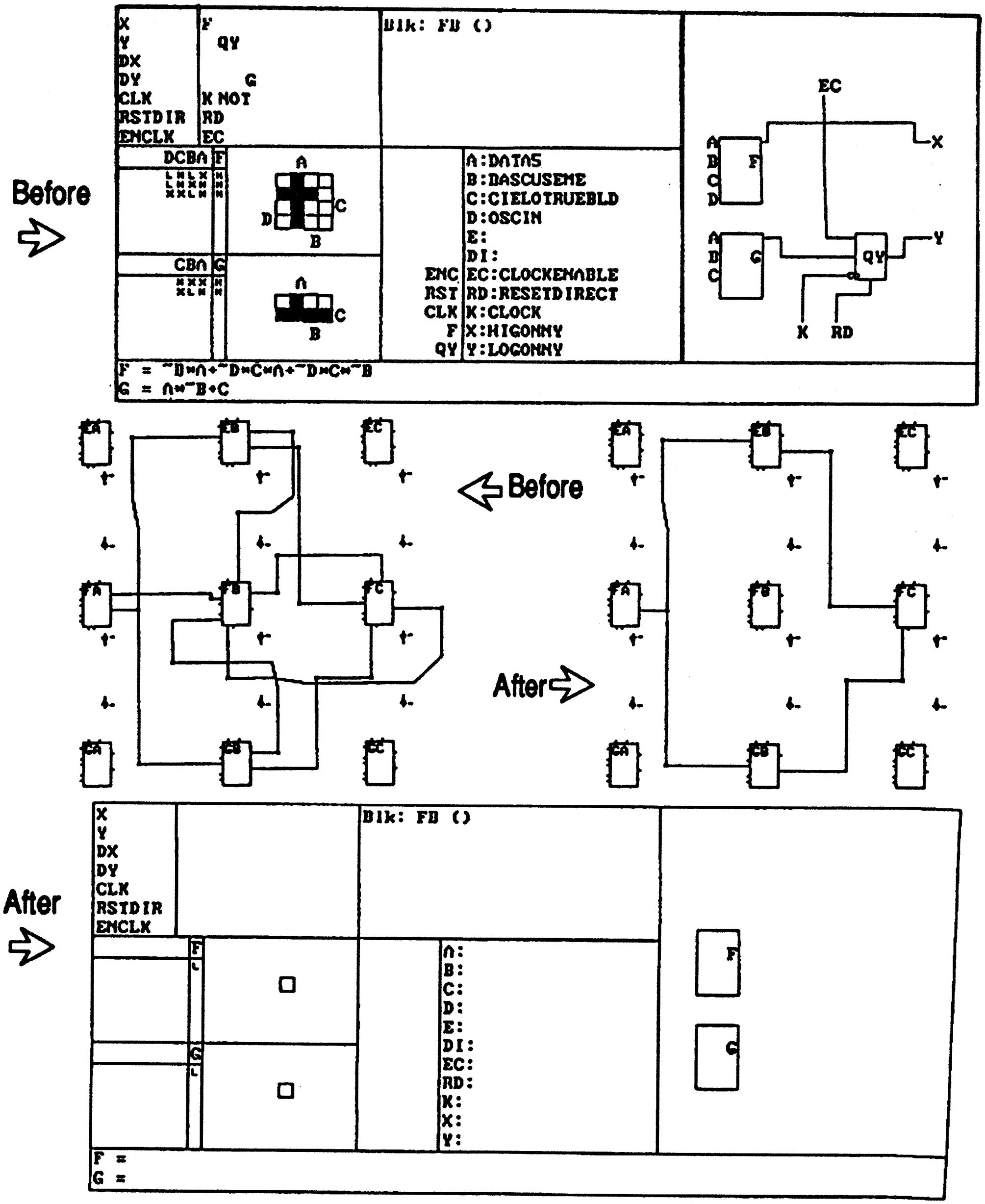
Delblk — Deconfigure a Block and Delete Pins From Nets

Menu	Blk
Syntax	delblk *block*
Abbreviations	delb, db
See also	Delpin, Clearblk

This command is a combination of Clearblk and Delpin for all pins of the block that belong to nets. It removes any assigned block name and deconfigures any spurs on the pins nets.

Figure 3-28 shows the results of entering this syntax:

```
delblk fb
```



```
X        F
Y          QY
DX
DY              G
CLK      K NOT
RSTDIR   RD
ENCLX    EC

Blk: FB ()

DCBA F
LHLX H
LHXH H
XXLH H

CBA G
HXX H
XLH H

      A:DATA5
      B:DASCUSEME
      C:CIELOTRUEBLD
      D:OSCIN
      E:
      DI:
ENC   EC:CLOCKENABLE
RST   RD:RESETDIRECT
CLK   K:CLOCK
F     X:HIGONNY
QY    Y:LOGONNY

F = ~D*A+~D*C*A+~D*C*~B
G = A*~B+C
```

```
X
Y
DX
DY
CLK
RSTDIR
ENCLX

Blk: FB ()

F
L
G
L

A:
B:
C:
D:
E:
DI:
EC:
RD:
K:
X:
Y:

F =
G =
```

Figure 3-28 Delblk Example

Delnet — Delete a Net

Menu	Net
Syntax	delnet *net*
Abbreviations	deln, dn
See also	Route, Addnet, Delpin

This command deletes the specified net and the pins that were connected to the net (using Delpin), and deconfigures its interconnect.

Delpin — Delete a Pin

Menu	Pin
Syntax	delpin *pin*
Abbreviations	delp, dp
See also	Unroute pin, Addpin, Routepin

This command removes the specified pin from its net and deconfigures its interconnect. It also deconfigures any spurs on the pin nets. To deconfigure a pin interconnect without deleting the pin itself, use the Unroute pin command.

Figure 3-29 shows the effect of entering the following string.

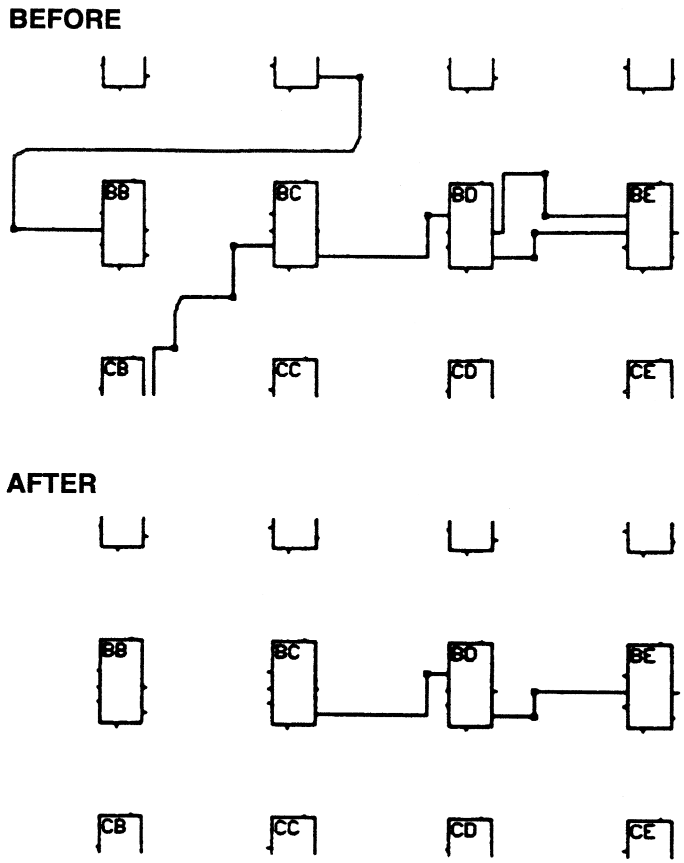
```
delpin bb.c bc.k bd.x
```

DOS — Exit XDE Temporarily and Enter DOS (PC only)

Menu	Misc
Syntax	dos
Abbreviations	none

This command exits XDE and brings up the DOS operating system prompt. Typing exit and pressing ↵ brings back the Design Editor in the same state it was in when you entered the DOS command.

Use this command for creating text files, printing files, and performing other functions that the Editor does not provide directly.



BEFORE

AFTER

Figure 3-29 Delpin Example

Download — Download Bitstream in Memory to Download Cable

Menu	Probe
Syntax	download
Abbreviations	down
See also	MakeBits, Debugload, Port

This command downloads the configuration bitstream in memory through the download cable. It is exactly the same as the Download command in the MakeBits program. Use it in XDE with the MakeBits command to create and download bitstreams through the download cable. (You can also perform these two actions using the Debugload command.) Before the download can take place, use the Port command to select a port to download to. If the port selected is User, the Download command writes a RBT (rawbits) file to the current directory and then executes a user-written program. The name of the RBT file is passed to the user program, which can then process and download the RBT file. For more information on the Download command, consult "The MakeBits Program" chapter in Volume 2 of the *XACT Reference Guide.*

NOTE *Download is applicable only to XC2000 and XC3000 designs. You must use XChecker for XC4000 designs.*

Draw — Create a Printable File of Design Information

Menu	Screen
Syntax	draw *filename format_type* world|block *block*
Abbreviations	none
See also	Print, Printer, Printpic (XDE Executive)

You can draw either the world or an individual block to a print file with the Draw command. You must enter the file name from the keyboard. If the file already exists, it is replaced by the new file.

The valid options for *format_type* are PostScript, PostScriptB, HPPlot_BW, HPPlot_Color, PMF.

If you choose the world option, Draw creates a file that contains the entire design. If you choose the block option, you must enter a block, and Draw creates a file that contains the Block Editor display of the specified block. In either case, this file will have the format *format_type*.

DRC — Start the Design Rules Checker

Menu	Misc
Syntax	drc *option*
Abbreviations	none
See also	Report

This command starts the Design Rules Checker (DRC) for the current design state. DRC checks the design for design rule violations. The check results display on the screen as the check progresses. To list the results in a file rather than on the screen, use the Report command. You can also start DRC from the XDE Executive. Table 3-14 lists the valid DRC Command options.

Table 3-14 DRC Command Options

Option	Characteristic
Verbose	Provides a running commentary of progress.
Noroute	Forces DRC to not check routing. Useful for checking unrouted designs.
Block *block* ...	Checks only the specified blocks.
Noblock	Causes all block checking to be skipped.
Net *net* ...	Checks only the specified nets.
Nonet	Causes all net checking to be skipped.
Informational	Alerts you to exiting subtle design conditions.

Editblk — Start the Block Editor

Menu	Blk
Syntax	editblk *block*
Abbreviations	editb, eb
See also	Endblk, Blk, Pie, Switch

This command makes the specified block the current block; all block editing commands affect this block. The Block Editor display for this block replaces the PIE display. The block remains the current block until you either enter the Endblk command or the Editblk command specifying another block.

Editblkcmt — Edit the Comment Text for a Block

Menu	Blk
Syntax	editblkcmt *block*
Abbreviations	editblkc, ebc
See also	Cmtblk

The Cmtblk command and the comment text associated with the specified block are presented on the XDE command line for editing.

Editeq — Edit Equation on Command Line in Block Editor

Menu	Config
Syntax	editq F | G | H
Abbreviations	ee
See also	Equate

You can only use the Editeq command when a current block is defined. The equation for the specified function is placed on the command line. To specify which equation to edit, either type the name of the function (for example, F), or use the mouse to select the equation for that function. For example,

```
editeq f
```

puts the equation for F on the command line like this:

```
F = A + B + C
```

The effect is the same as if you had typed the equation at the command line. You can use the line-editing features to alter the equation as desired.

You can edit equations without using the Editeq command simply by using the mouse to select the equation on the Block Editor display. The effect is the same as if an Editeq command had been issued for that equation; that is, the equation is copied to the command line for editing.

Editnet — Alter the Routing of a Net

Menu	Net
Syntax	editnet net PIP
Abbreviations	editn, en
See also	Addnet, Route, DRC, Autodrc, Show Directional

You can toggle a PIP, toggle a switching matrix connection, or route a net manually with this command.

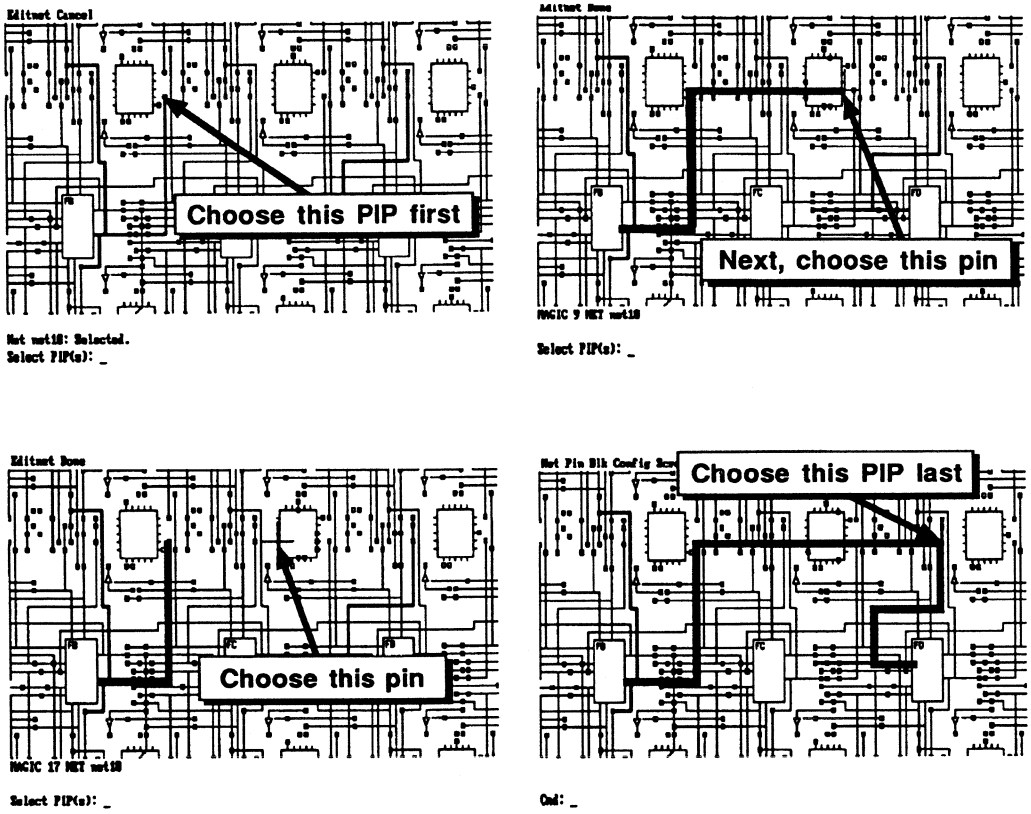
Toggle a PIP

Use the mouse to select the PIP you want to toggle on or off. When a PIP is on, the associated interconnect are connected. PIPs associated with pins not on the specified net cannot be altered; they must be either associated with this net or not associated with any net or pins.

Toggle a Switching Matrix Connection

If you enter a switching matrix pin, the next pin you specify must be one of the other pins in the same switching matrix. Switching-matrix pins are entered in pairs. When entered, if two pins in a pair are connected, they are disconnected. When entered, if they are disconnected, they are then connected. Some switching matrix connections are illegal; these restrictions depend on the location of the switch matrix and the type of LCA.

Figure 3-30 illustrates the effect of selecting PIPs and switching-matrix pins using Editnet.



```
Editnet Cancel
Net net18: Selected.
Select PIP(s): _
```

```
Editnet Done
MAGIC 9 NET net18
Select PIP(s): _
```

```
Editnet Done
MAGIC 17 NET net18
Select PIP(s): _
```

```
Net Pin Blk Config Scr
Cmd: _
```

Figure 3-30 Editnet Example

WARNING!!!

Some programmable interconnect points (PIPs) are not simple pass transistors, and using Editnet can introduce design errors. Always start DRC after using Editnet, or have Autodrc enabled.

When Autodrc is enabled, XDE checks PIPs for proper use as they are programmed, so that you can quickly catch many potentially subtle errors. For example, it is not possible to use two PIPs on a block input as general interconnect, where one is programmed to drive into the block input, and the other is programmed to drive out into another channel.

Block inputs (the row of PIPs connected to the input pin) are inputs of a multiplexer, not simple pass transistors. Therefore, you can only program a single PIP on a block input. If you try to use the block-input line as a routing resource, the activation of those inputs results in a net that appears routed, but DRC and the timing calculator indicate that the route is not complete. The block input PIPs are unidirectional multiplexer inputs, and the use of more than one of those PIPs on a block input is illegal.

In the XC3000 series, there are sets of special PIPs adjacent to switching matrices. Refer to Figure 3-16 for an illustration of the XC3000-series structure.

- In the XC3000 architecture, two pairs of PIPs found above and below the switching matrix are associated with the horizontal Longlines. The PIP vertically aligned with the TBUF input PIP is a buffered access from the local vertical interconnect to the horizontal Longline. It is identified by buffered V->H on the status line when located by the cursor. The other provides access from the horizontal Longline to the vertical interconnect and is identified as H->V when located by the cursor. These PIPs can only be used to connect signals in the direction indicated (V->H or H->V).
- Also in the XC3000 LCA, additional special BIDI PIPs are located directly adjacent to (above or to the right of) the switching matrix, except at the die perimeter. Use the ShowBIDI command to show the location of the bidirectional-buffer PIPs. These represent bidirectional (BIDI) buffers for signal restoration and load isolation. The software automatically selects the direction on the basis of the location of the network source.

Most PIPs for the XC2000, XC3000, and XC4000 LCA devices have directional constraints. These PIPs can only be driven in certain ways; misprograming them violates architectural rules. When you point the cursor at a PIP from within the Design Editor, the status line at the bottom of the screen reports the characteristics of that PIP. Table 3-15 lists the valid PIP characteristics.

Table 3-15 PIP Characteristics

PIP attribute	Description
Buffered	A PIP with drive capability
P0 or P1	P0 indicates that the PIP is not programmed; P1 indicates that the PIP is programmed
ND	Non–directional PIP, can drive in any direction
D:	Directional PIP, certain rules must be followed
H–>V	Horizontal to Vertical, indicates that driving vertical to horizontal through this PIP is a violation
V–>H	Vertical to Horizontal, indicates that driving horizontal to vertical through this PIP is a violation
T–>C	Tee to Cross, indicates that driving from the cross to the stem of the "T" through this PIP is a violation
C–>T	Cross to Tee, indicates that driving from the stem of the "T" to the cross through this PIP is a violation
CW	Clockwise, indicates that driving counter–clockwise through this PIP is a violation
CCW	Counter–Clockwise, indicates that driving clockwise through this PIP is a violation

As an example: consider a PIP buffered P1 D:V->H. This indicates that the PIP is buffered, it is currently turned on, and it cannot be driven horizontal to vertical. It could be a PIP that drives onto a horizontal Longline. You can drive a signal onto the Longline and/or pass through the PIP vertically, but you cannot drive a signal being sourced from the Longline out into the switch matrices from this PIP.

Endblk — Deselect the Current Block

Menu	Blk
Syntax	endblk
Abbreviations	end, endb
See also	Editblk

This command deselects the current block and returns it to its normal color in the PIE display. If you are in the Block Editor display, Endblk returns you to the PIE display.

Endfile — Exit from a Command File

Menu	none
Syntax	endfile
Abbreviations	none
See also	Execute

You can only use the Endfile command in command files. It does not appear on any menu.

A command file is a text file of Design Editor commands. When the Editor is performing commands from a command file and encounters the Endfile command, it exits from the command file and enters its normal mode of receiving commands from the keyboard or mouse.

Enprtsc — Enable the Prtsc Key (PC only)

Menu	none
Syntax	enprtsc
Abbreviations	none
See also	Print

You can only enter this command from the keyboard. It does not appear on any menu.

The Prtsc key (shifted) normally sends the information displayed on the screen to the printer. When you start the Design Editor, this function is disabled. The Enprtsc command restores the screen printing function. Once you enable this command, there is no way to disable it.

You should start the Graphics driver by entering the Graphics command in DOS prior to starting XDE to ensure that graphics screen images are correctly output to the printer.

Enprtsc does not support graphic output in the EGA display mode. Text output in EGA mode is supported.

WARNING!!! *If no printer is attached to the system and the Prtsc (shifted) key is pressed, the PC stops working until the system is rebooted (by pressing the Ctrl, Alt, and DEL keys simultaneously). This problem only occurs if you use the Enprtsc command.*

Eqnetnames — Display Editblock Equations User Net/CLB Pin Names

Menu	Profile
Syntax	eqnetnames on|off
Abbreviations	eqn

While Eqnetnames is on, you can enter equations and display them in the Block Editor and Queryblk commands in terms of names assigned to the nets. For example, you can enter:

```
f =~ reset * (data0 * sel + data1 * ~ sel)
```

instead of:

```
f = ~a * (b * c + d * ~c)
```

Equate — Configure a CLB Logic Function

Menu	Config
Syntax	equate *tag* = *boolean_expression*
Abbreviations	eq
See also	Config, Order, Clear

Both the Equate and the "=" are optional. The tag can be F, G, or H, depending on the LCA family and the base configuration. The Boolean expression can be a logic function of up to three inputs (XC2000 base FG), four inputs (XC2000 base F, XC3000 base FG and FGM, XC4000 base FG), or five inputs (XC3000 base F). You can use the mouse in EditBlk to select an existing equation, which you can then modify using the keyboard. The logic operators follow in order of decreasing precedence.

Table 3-16 Logic Operators

Symbol	Function
~	NOT
*	AND
@	XOR
+	OR

NOTE *AND and XOR have the same precedence and are evaluated right to left.*

You can use parentheses to group subexpressions.

When you enter an equation, the truth table and Karnaugh map are automatically adjusted. Conversely, you can toggle Karnaugh map values by selecting them with the mouse, and the equation is automatically adjusted.

An alternative to entering function equations through the keyboard with the Equate command is entering them using only Karnaugh maps specified through the Order or Config commands.

Figure 3-31 shows the results of entering the equation F = A + B * C.

Execute — Perform Commands from a Command File

Menu	Execute
Syntax	execute *filename parameter*
Abbreviations	exec
See also	Endfile, Macro

A command file is a text file containing Design Editor commands. Execute causes the Editor to perform the commands in the order in which they appear in the file. When the end of the file or the Endfile command is encountered, the Editor can again receive command input from the keyboard or mouse.

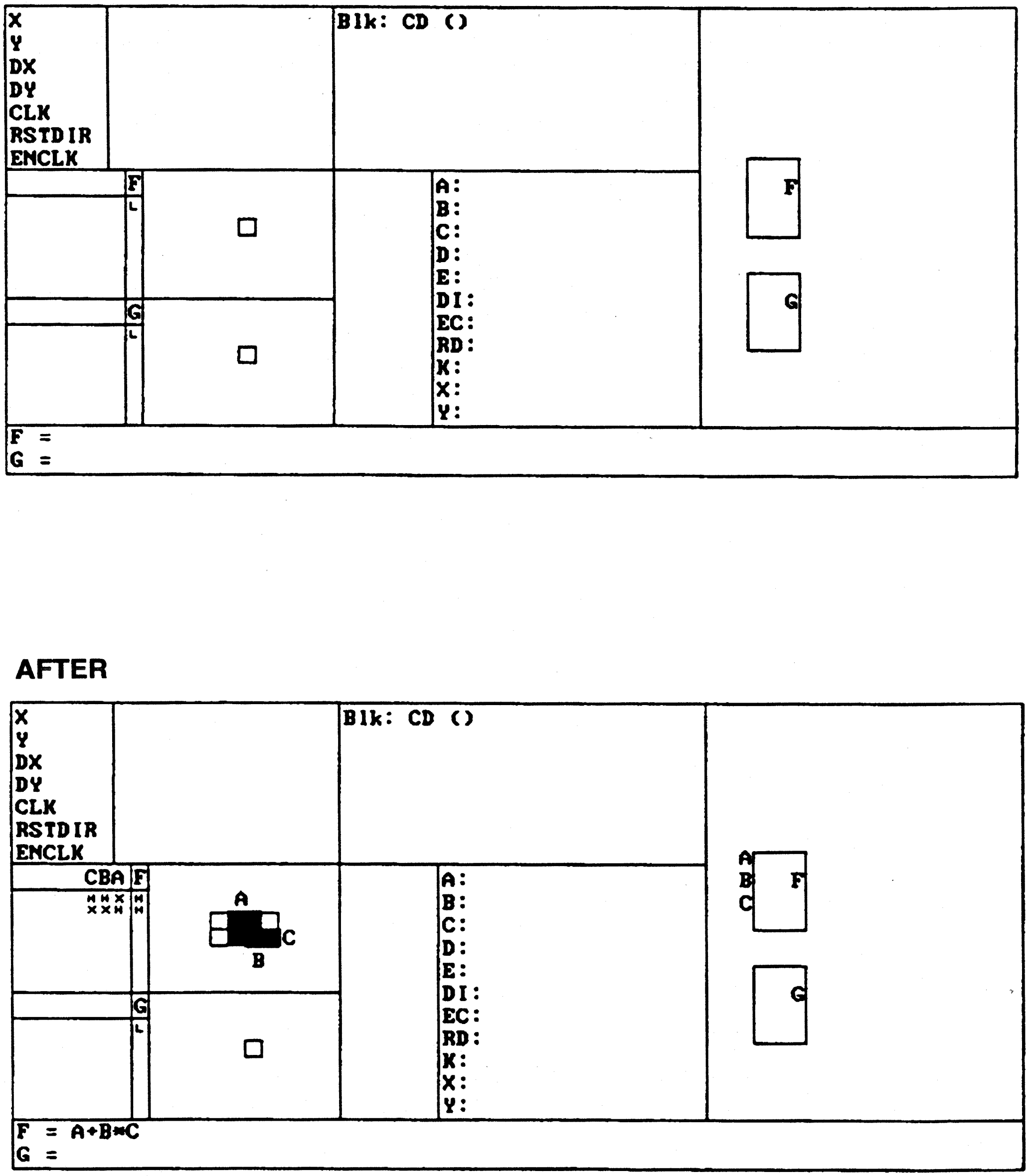
If a command file contains an Execute command, a second command file is performed within the first. Up to eight command files can be open at the same time.

If you use wildcards in a command file, the ARE YOU SURE? prompt is not used; you do not need to include a response for it.

You can include variable parameters at any place in a command file. The expression %*n*, where *n* is any number, represents a variable parameter. When the Execute command is performed, the specified parameters replaces the %*n* expressions. The order in which the parameters are given corresponds to the expressions they replace; the first parameter replaces the expression %1 wherever it appears in the command file, the second parameter replaces %2, and so on.
Figure 3-32 shows an example of parameter substitution.

Unlike the Paste and Macro commands, the Execute command performs no processing of variable parameters, even if parameter declarations exist in the command file.

NOTE *An Exit command is not permitted in Execute files.*



BEFORE

```
F =
G =
```

AFTER

```
F = A+B*C
G =
```

Figure 3-31 Equate Example

Command File Test Text Example:

```
EDITBLK %1
BASE F
EQUATE F (A*B) + (C@D)
CONFIG X:F
ENBLK
ADDPIN %2 %1.A
ADDPIN %3 %1.X
```

Input Commands

```
EXECUTE TEST GC NET1 NET4
```

Commands Performed

```
EDITBLK GC
BASE F
EQUATE F (A*B) + (C@D)
CONFIG X:F
ENDBLK
ADDPIN NET1 GC.A
ADDPIN NET4 GC.X
```

Figure 3-32 Command-File Parameter Substitution

Exit — Leave Design Editor and Do Not Save Design State

Menu	Misc
Syntax	exit
Abbreviations	none
See also	File, Save

This command ends the current editing session and returns you to the XDE Executive. The design is not saved. You can save the design from the XDE Executive or from the Editor, using the Save or File command.

File — Save Design State and Leave the Design Editor

Menu	Misc
Syntax	file *filename*
Abbreviations	none
See also	Save, Quit

This command saves the current design state to the specified design file and returns you to the XDE Executive. File prompts you before overwriting an existing file, if you specify a name other than the current file name. If you do not specify a file, the name of the current design file name is used. The Editor automatically adds the LCA extension to the file name if you do not.

Since you can leave MakeBits with a design in the tied state, File or Save issues a warning and prompts for a response if this is the case. This warning mitigates the possibility of overwriting an LCA file with a Tied version of the design.

Use Save to save the current design state without leaving the Editor. Use Exit to leave the Editor without saving the current design state.

If the design cannot be saved, the Editor issues a message and the editing session continues.

Find — Move the Cursor and Window to a Location

Menu	Screen
Syntax	find *location*
Abbreviations	fi
See also	Findblk, Findnet, Findpin

This command moves the cursor to the specified location and adjusts the window to center the cursor in the display, if possible. The location can be a block, pin, PIP, switching matrix pin, grid location, or net. If you specify a net, Find moves the cursor to its source pin. If the net has no source pin, Find moves the cursor to the first load pin.

You can quickly move the cursor to a location outside the window by using the mouse to select the desired location in the world view.

Findblk — Move the Cursor and Window to a Block

Menu	Screen
Syntax	findblk *block1* [*block2*] [*block3*]
Abbreviations	findb
See also	Find, Findnet, Findpin

Findblk works similarly to the Find command, except it specifically finds blocks. After you specify a block, Findblk moves the cursor so that it points at the block and adjusts the display window so that the cursor is in the center, if possible. Unlike the Find command, Findblk can find multiple blocks. You can specify several blocks, and Findblk locates the first, then prompts you to hit a key before it attempts to find the next block. This command is useful when you are determining the relative position of each block in a set.

Menu	Screen
Syntax	findnet *net1 [net2] [net3]*
Abbreviations	findn
See also	Find, Findblk, Findpin

Findnet works similarly to the Find command, except it specifically finds nets. After you specify a net, Findnet moves the cursor so that it points at the source of the net and adjusts the display window so that the cursor is in the center, if possible. Unlike the Find command, Findnet can find multiple nets. You can specify several nets, and Findnet locates the source of the first, then prompts you to hit a key before it attempts to find the next net. This command is useful when you are determining the relative position of each net in a set.

If the specified net has no source pin, Findnet moves the cursor to the first load pin.

Findpin — Move the Cursor and Window to a Pin

Menu	Screen
Syntax	findpin *pin1* [*pin2*] [*pin3*]
Abbreviations	findp
See also	Find, Findblk, Findnet

Findpin works similarly to the Find command, except it specifically finds pins. After you specify a pin, Findpin moves the cursor so that it points at the pin and adjusts the display window so that the cursor is in the center, if possible. Unlike the Find command, Findpin can find multiple pins. You can specify several pins, and Findpin finds the location of the first, then prompts you to hit a key before it attempts to find the location of the next pin. This command is useful when you are determining the relative position of each pin on a net.

Flagblk — Flag Blocks for Use with the Path Delay Calculator

Menu	Timing
Syntax	flagblk *option block*
Abbreviations	none
See also	Queryblk

This option allows you to flag certain blocks that are then given special consideration by the path-delay calculator. The flags are listed below. After selecting a flag, a menu appears listing valid blocks. Information from Flagblk is stored in the MRG files created by the SaveMargins command.

IOB_Enable_O_I Trace Paths From the O Pin to the I Pin

IOB_Enable_O_I allows paths from the IOB O pin to the pad and back into the chip through the IOB I pin. Use this option to control tracing through bidirectional IOBs, which can cause false paths.

IOB_Disable_O_I Disallow Paths from the O Pin to the I Pin

IOB_Disable_O_I disallows paths from the IOB O pin to the pad and back into the chip through the IOB I pin. This is the default.

IOB_Enable_T_I Allow Paths from the T Pin to the I Pin

IOB_Enable_T_I allows paths from the IOB T pin to the pad and back into the chip through the IOB I pin. Use this option to control tracing through bidirectional IOBs, which can cause false paths.

IOB_Disable_T_I Disallow Paths from the T Pin to the I Pin

IOB_Disable_T_I disallows paths from the IOB T pin to the pad and back into the chip through the IOB I pin. This flag is the default.

TBUF_Enable_I_O
Allow Paths From the I Pin to the O Pin

TBUF_Enable_I_O allows paths that go through the TBUF I pin to the TBUF O pin. This is the default.

TBUF_Disable_I_O
Disallow Paths from the I Pin to the O Pin

TBUF_Disable_I_O disallows paths from a TBUF I pin to the TBUF O pin.

TBUF_Enable_T_O
Allow Paths from the T Pin to the O Pin

TBUF_Enable_T_O allows paths that go through a TBUF T pin to the TBUF O pin. This flag is the default.

TBUF_Disable_T_O
Disallow Paths from the T Pin to the O Pin

TBUF_Disable_T_O disallows paths through a TBUF T pin to the TBUF O pin.

CLB_Enable_WE
Allow Paths through Write-Enabled Input of CLB RAM

CLB_Enable_WE 164allows paths through the write-enabled input of CLB RAM. This flag is the default.

CLB_Disable_WE
Disallow Paths through Write-Enabled Input of CLB RAM

CLB_Disable_WE disallows paths through the write-enabled input of CLB RAM.

CLB_Enable_DIN
Allow Paths through Data Input(s) of CLB RAM

CLB_Enable_DIN allows paths through Data inputs of CLB RAM. This flag is the default.

CLB_Disable_DIN
Disallow Paths through Data Input(s) of CLB RAM

CLB_Disable_DIN disallows paths through Data inputs of a CLB RAM.

CLB_Enable_SR_Q
Allow Paths from Async Set/Reset Input to the Q Output

CLB_Enable_SR_Q allows paths from a CLB Asynchronous Set or Reset input, to the Q output of the affected flip-flop. This flag is the default for Asynchronous Set/Reset paths.

CLB_Disable_SR_Q
Disallow Paths from Async Set/Reset Input to the Q Output

CLB_Enable_SR_Q disallows paths from a CLB Asynchronous Set or Reset input to the Q output of the affected flip-flop.

FlagIOB — Flag an IO Block for Internal/External Use (XC2000 and XC3000 only)

Menu	Aprcon
Syntax	flagiob *internal/external blk*
Abbreviations	none
See also	Markblock, Marknet, Markpin, Weightnet

By flagging an IOB external and saving the information to a constraints file that you can use with APR, the IOB is placed on a bonded pad. Flag an IOB Internal if you want the IOB to be placed on an unbonded pad. The named blocks must be IOBs. To have effect, any flags you set within the Aprcon menu must be saved to disk using the Writecst command and then used with APR.

Flagnet — Flag Specified Net with an Attribute

Menu	Net
Syntax	flagnet *option net*
Abbreviations	`fn`

Flagnet can attribute Critical and Noncritical options to specified nets.

By default, all nets are Noncritical. Critical nets are protected from alteration during the tie-down option of bitstream generation.

Noncritical nets are not protected; you can use them to tie down interconnects, which could result in increased path delays.

Here are two examples:

The following syntax makes the myclock net and all bus* nets critical:

```
flagnet critical myclock bus*
```

The following syntax removes the critical attribute from the bus* nets:

```
flagnet noncritical bus*
```

The Critical option displays the specified Critical nets.

The Critical attribute within XDE is not related to the APR Critical net flags, which dictate net routing order. That is, the APR Critical flag and the XDE Critical flag are separate specifications used for separate purposes.

Help — Explain Design Editor Commands

Menu	Misc
Syntax	help
Abbreviations	none

The Help command displays a menu of all the topics for which on–line help is available. While this menu displays, use the mouse to select a topic to get more information.

Help is also available at any time by pressing F1 when the mouse is over a menu item or in the EditLCA display. When the mouse is in the EditLCA screen, information about the LCA object under the cursor displays when you press F1.

Hilight — Draw Net Interconnects and Stubs in Color

Menu	Net
Syntax	highlight *color net*
Abbreviations	hi, hil
See also	Unhilight, Redraw, Updatehilight

This command redraws all of the net interconnects in the specified color. It also shows any stubs that the net uses that are not normally shown. If the net is not routed, the source and destination pins are drawn but remain unconfigured. The interconnects remain highlighted until they are redrawn by other commands such as, Route. The Unhilight command is the inverse of Hilight. Figure 3-33 shows the effect of Hilight.

Hilightwide — Change the Width of the Highlight Lines

Menu	Profile
Syntax	highlightwide
Abbreviations	none
See Also:	Hilight

This command causes the Hilight command to draw the highlighted nets using wider lines, which causes them to stand out better on printouts and monochrome screens.

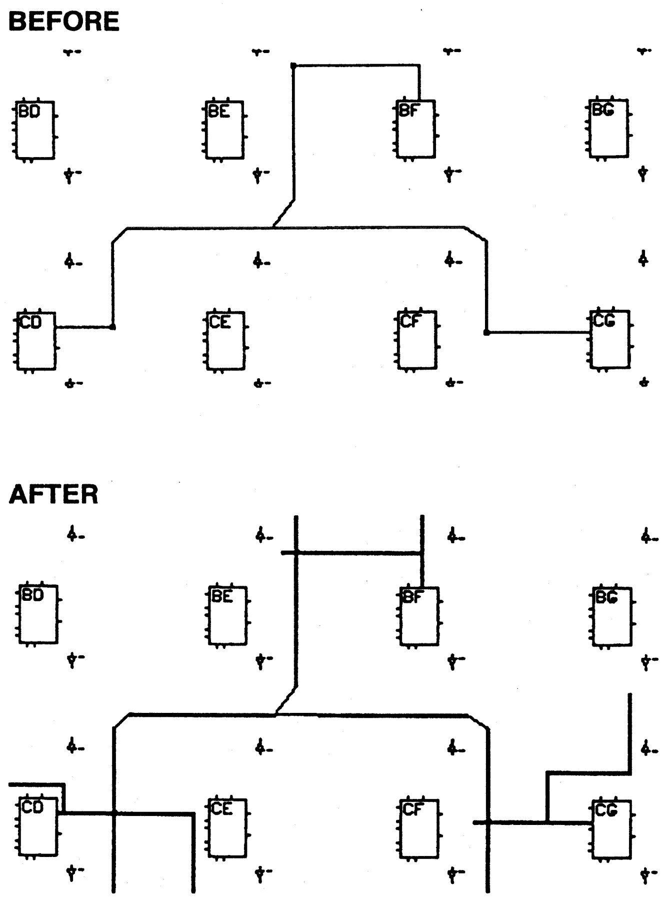


Figure 3-33 Hilight Example

Joinnet — Merge Nets Together and Deletes Secondary Net

Menu	Net
Syntax	joinnet *net1 net2*
Abbreviations	joinn, jn
See also	Addnet, Delnet, Route, Autoroute

Joinnet adds all of the pins on net2 and any other specified secondary nets to net1. The secondary nets are deleted and the primary net is rerouted if possible when Autoroute is on.

Keydef — Define a Function Key

Menu	Profile
Syntax	keydef *keyname_definition*
Abbreviations	ke

The key name must be one of the PC 12 function keys F1 through F12. You can use the Shift, Ctrl, or Alt key in combination with a function key; enter Shift, Ctrl, or Alt before entering the function key name.

The keystrokes you type as the function key definition (terminated by enter or "\") are stored under that function key; whenever you press the function key, the definition is typed automatically. Use function keys to store command sequences you use frequently.

For example, if you enter the following syntax:

```
keydef F10 print display print1
```

then pressing the F10 key becomes the same as typing:

```
print display print1
```

You can terminate a function key definition by pressing ↵ or the backslash (\ ). If you terminate the keystroke sequence by pressing ↵, the carriage return is included in the key definition. If you terminate a key definition, the command defined by the key displays and executes immediately when you press the key. If the definition is terminated with a backslash (\ ), the key definition displays, but does not execute when you press the key. You have to press ↵ to execute commands so defined.

To delete a function key definition, enter Keydef and the key name and press ↵ without typing a definition.

The Editor has its own function key settings independent of the XDE Executive or any other subprogram. Initial settings for the Editor function keys are contained in the editlca.pro file. Use Saveprofile to save changes to function key definitions in the editlca.pro file.

Macro — Perform Commands from a Macro File

Menu	Misc
Syntax	macro *filename parameter*
Abbreviations	mac
See also	Cut, Cutmacro, Paste, Execute

Macros are a shorthand method for storing configuration information. Entering configurations from macros saves you the time and effort of manually entering the configuration.

A macro file is a command file with parameter declarations. See the Execute command for information on command files. Parameter declarations, which are always located at the beginning of the file before any commands, perform two functions.

- Prompts for and receives parameter values
- Checks validity of received values

Macros offer two advantages over command files. First, there is no need to remember the parameter type or order required by the macro; it automatically prompts you for them in the correct order. Second, a macro is not performed if any of your parameters are not valid for the commands for which they are used. Without parameter checking, the validity of a parameter is only tested as the command is encountered. If a command is not performed because of an invalid parameter, the command file is not completely exectued.

All macro files must have the .mac extension. If you do not add the .mac extension, the Editor automatically adds it before invoking the macro file. If the MAC file name already exists, the Macro command renames it with the .omf extension, displays a message that it has renamed the file, and then creates the new MAC file.

Figure 3-34 shows an example of a short macro, the command to start it, and the resulting configuration.

```
Parameter NAME ? Enter instance name:
Parameter NET net1 Select net1 net:
Parameter NET net0 Select net0 net:
Parameter CLB ? Select AC block:
Editblk %4
Base FG
Config X:F Y: F:A:B G: Q: SET: RES: CLK:
Equate F = A+~B
Endblk
Addpin %2 %4.A
Addpin %3 %4.B
Addpin %lnet2 %4.X
```

```
COMMAND: MACRO macfile test tnet1 tnet2 AC
```

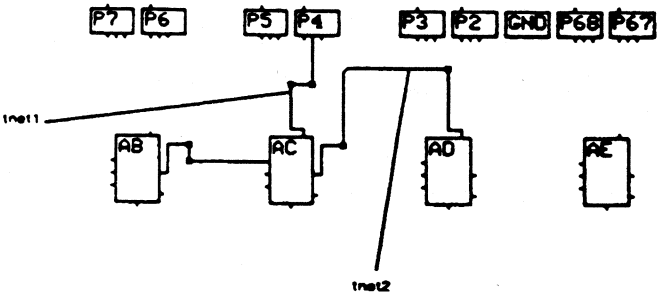


Figure 3-34 Macro Example

MakeBits — Create a Bitstream of the LCA File in Memory

Menu	Probe
Syntax	makebits
Abbreviations	makeb
See also	Download, Debugload

This command is the same as the MakeBits command found in the MakeBits program. MakeBits creates a configuration bitstream on the basis of the current LCA design. Use MakeBits in XDE with the Download command to create and download bitstreams through the download cable. (You can use Debugload to create and download bitstreams.) For more information on the MakeBits command, consult "The MakeBits Program" chapter in Volume 2 of the *XACT Reference Guide*.

Markblk — Set Constraint File Flag for a CLB or IOB (XC2000 and XC3000 only)

Menu	Aprcon
Syntax	markblock *option block*
Abbreviations	markb
See also	Marknet, Markpin, FlagIOB, Weightnet

This command sets one or more of the following constraint flags for placing a XC2000/XC3000 block: Locked, Unlocked, Prohibited, and Notprohibited. See "The Automatic Place and Route Program" chapter in Volume 2 of the *XACT Developement System Reference Guide* for more information about APR and constraint files.

Marknet — Set Constraint File Flag for a Net (XC2000 and XC3000 only)

Menu	Aprcon
Syntax	marknet *option net*
Abbreviations	markn
See also	Markblk, Markpin, FlagIOB, Weightnet

This command sets one or more of the following constraint flags: Critical, Uncritical, Longline, NoLongline, Normal, Locked, and Unlocked. See "The Automatic Place and Route Program" chapter in Volume 2 of the *XACT Developement System Reference Guide* for more information about APR and constraint files.

Markpin — Set Constraint File Flag for a Pin (XC2000 and XC3000 only)

Menu	Aprcon
Syntax	markpin *option pin*
Abbreviations	markp
See also	Markblk, Marknet, FlagIOB, Weightnet

This command marks a pin as either locked or unlocked. See "The Automatic Place and Route Program" chapter in Volume 2 of the *XACT Reference Guide* for more information about APR and constraint files.

Mouse — Change the Mouse Configuration

Menu	Profile
Syntax	mouse *option function*
Abbreviations	mou
See also	Cursor, Switch

Mouse allows you to define the mouse-button functions.

There are three options (B1, B2, or B3) that define mouse button functions, which you determine by selecting both mouse button and function.

- B1 selects button #1.
- B2 selects button #2.
- B3 selects button #3 (has no effect if there is no third button).

The left button of most 2-button mouse buttons equals B1; the right button is B3, and both buttons pressed at once equals B2.

You must also enter one of four functions to determine the action of the selected button. These functions are described below.

- Select enters the cursor location. This is the usual setting for all the mouse buttons. You can indicate block, net, or pin selections by positioning the cursor and clicking the mouse button corresponding to Select.
- Done applies the newly defined function and enters the Done command.
- Menu displays the most recent menu and moves the cursor to the most recent selection. If a menu is already displayed, the mouse button enters the cursor location (same as Select).
- Switch performs the Switch function by switching between the PIE display and the current-block Block Editor.

The XDE Executive has its own setting for the mouse port connection. Although you can set the Design Editor mouse configuration to be different from the mouse configuration in the XDE Executive, it is normally the same to avoid confusion. Use the Saveprofile command to save the Editor mouse configuration in the editlca.pro.

Moveblk — Move Configuration and Net Connections Between Blocks

Menu	Blk
Syntax	moveblk *blkl blk2*
Abbreviations	moveb, mb
See also	Copyblk, Swapblk, Swapblkgroup

This command deletes the internal configuration and net connections of *blkl* and copies them to *blk2*. If the Autoroute option is on, new interconnects are configured. If *blkl* has been renamed, that name is also moved to *blk2*. If the block nets contain any spurs, the spurs are deconfigured.

Blk2 must be unconfigured and its pins must not belong to any nets. The blocks must be either both CLBs or both IOBs.

Moveblk is similar to Copyblk except the source block is deleted using an equivalent of the Delblk command. Figure 3-35 illustrates the effect of entering the following command.

```
moveblk ad ae
```

Movepin — Move a Pin Net Connection to Another Pin

Menu	Pin
Syntax	movepin *pinl pin2*
Abbreviations	movep, mp
See also	Addpin, Routepin, Unroute pin, Delpin

This command removes *pinl* from the net, deconfigures its interconnect, and adds *pin2* to the *pinl* net, and, if the Autoroute option is on, configures the interconnect for *pin2*. If the net contains any spurs, they are deconfigured. *pinl* must belong to a net; *pin2* must not.

Figure 3-36 shows the result of entering the following command.

```
movepin gc.a gb.a
```

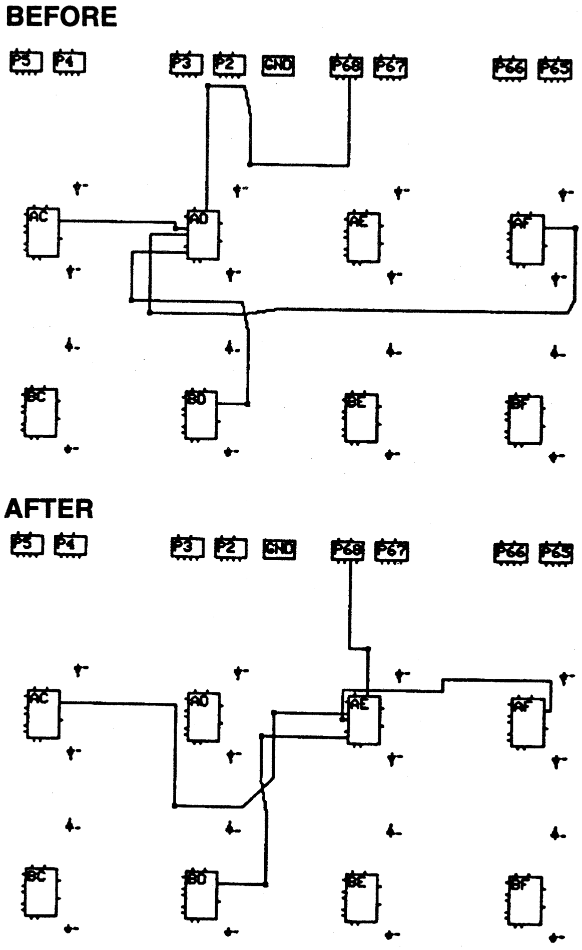


Figure 3-35 Moveblk Example with Autoroute On

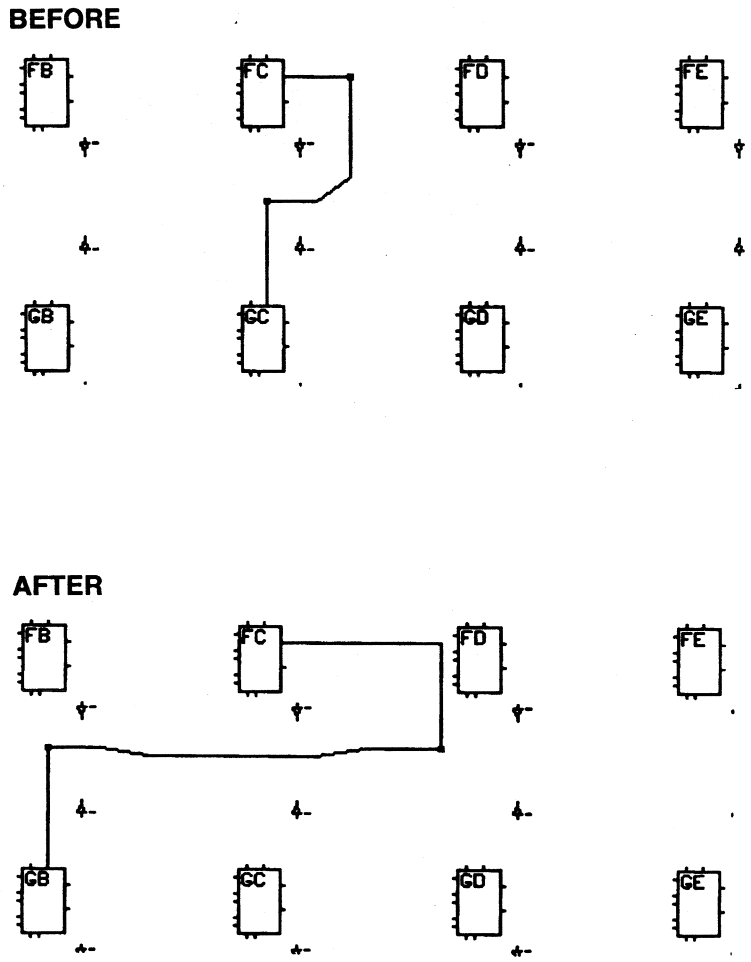


Figure 3-36 Movepin Example (with Autoroute on)

Nameblk — Assign a Name to a Block

Menu	Blk
Syntax	nameblk *nameblk*
Abbreviations	nameb, nb

This command gives the specified name to a block. The new name replaces the old name on the PIE display. If the new name is absent, any name previously given to the block is removed.

When you enter commands, you can use either the new name or the block locational name (physical name) to specify the block.

To delete the new name, enter the Nameblk command and the block name, then press ↵ without entering a new name.

Block Name Restrictions

- Names cannot be more than 1023 characters long.
- Names must consist only of letters, numbers, or the characters _ and $. The Editor is not case sensitive.
- Names cannot begin with a number.
- Names must be unique. The Design Editor already uses the following names.

Name	Description
AA through HH	Block–location names (XC2000)
AA through TP	Block–location names (XC3000)
CLB_RnCn	Block–location names (XC4000). For example: CLB_R3C2
P*nnn*	External pin names (PLC packages, PG packages have other letters.
PADnnn	Pad names
A, B, C, C1, C2, C3, C4, D, E, F1, F2, F3, F4, G1, G2, G3, G4, K, EC, DI, RD, X, XQ, Y and YQ	CLB pins
I, I1, I2, IQ, O, T, IK,and OK	IOB pins

NOTE *nnn equals any number from 1 to 999.*

NOTE *Additional block names might be reserved, depending on the type of LCA that is being configured.*

Namenet — Rename a Net

Menu	Net
Syntax	namenet *oldnet newname*
Abbreviations	namen, nn
See also	Addnet

Oldnet is the name of the existing net; *newname* is the new name you want the net to have. The same naming restrictions apply to nets as to blocks (see Nameblk).

NetMenus — Display Scrolling Menus of Nets

Menu	Screen, Profile
Syntax	netmenus on|off
Abbreviations	none

When the NetMenus option is enabled, a menu of net names scrolls when you choose commands that require net selection.

Order — Order CLB Logic Function Inputs

Menu	Config
Syntax	order *tag var*
Abbreviations	orde
See also	Equate, Config

The tag can be F or G if the base configuration is FG or FGM; otherwise, it must be F. H is also a valid tag in an XC4000 CLB. If you specify variables that are different from those already given for the tag, the tag logic equation is cleared; otherwise, the logic equation input order in the truth table and Karnaugh map is changed to match the specified order.

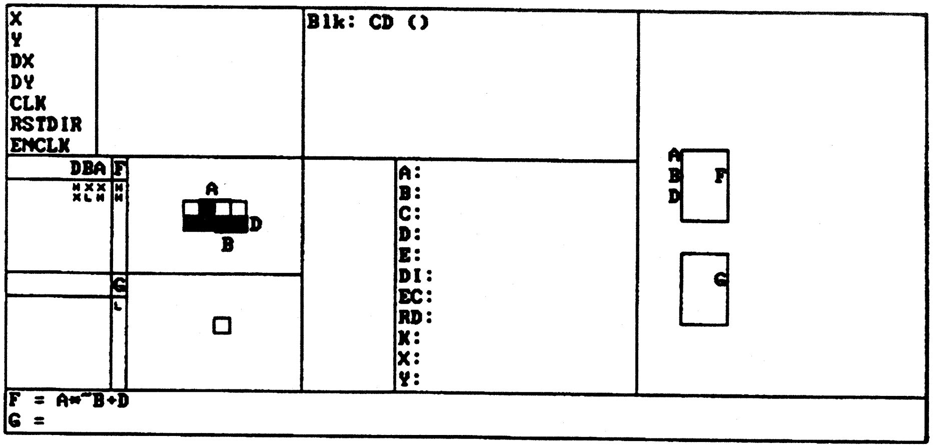
The Order command sets up a Karnaugh map for the selected variables. A filled-in cell indicates a 1; an empty cell indicates a 0. You can fill in the Karnaugh map using the mouse. To set a Karnaugh-map cell to 1, place the cursor at the cell and press the mouse button. Press the button again to set the cell to 0. With the Order command, you can enter a logic equation using only the mouse, set up the empty Karnaugh map, then fill in the Karnaugh-map cells.

If you specify Order with a tag but no variables, the currently used variables are ordered alphabetically. For example, if the CLB is configured with F = A + C + B, then the string, ‘order f a b c’ yields the same results as, ‘order f.’

Figure 3-37 shows the results of entering the following syntax.

```
order f d b a
```

BEFORE



AFTER

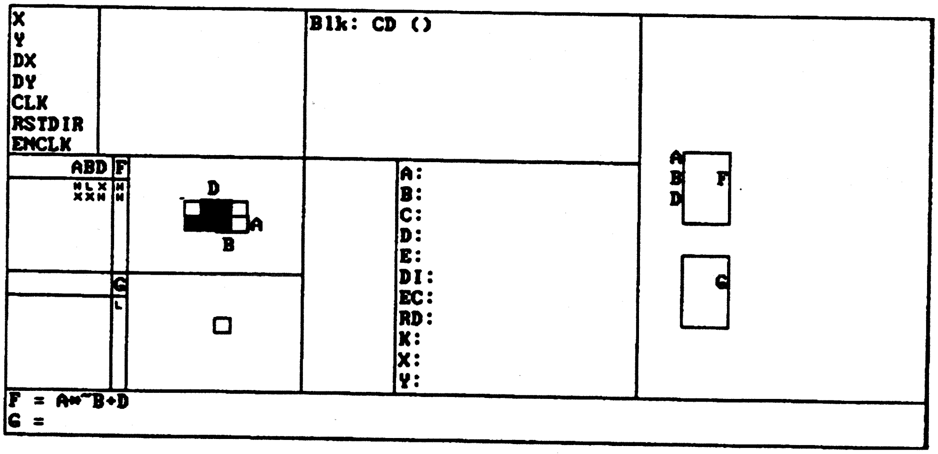


Figure 3-37 Order Example

Paste — Perform Commands from a Cut File

Menu	Misc
Syntax	paste *filename parameter*
Abbreviations	pas
See also	Cut, Cut Macro, Macro, Execute

The Paste command is identical to the Macro command, except that the file whose contents are executed must have the .cut extension. It is convenient to use the Paste command in conjunction with the Cut command to move a portion of a configuration from one area of an LCA design to another.

When you specify a block as a parameter value, Paste checks if it has already been specified. If so, its subsequent specification is rejected and Paste issues an error message.

Piecolors — Change Color of an Item in Editor Display

Menu	Screen
Syntax	piecolors *item color*
Abbreviations	piec

This command is only available in 8- or 16-color modes. Piecolors assigns colors to most EditLCA interconnect editor display features. You can change the parts of the PIE display listed in Table 3-17.

NOTE *If XDE finds a color that is not in the current palette, while loading a design, it uses the default color.*

NOTE *Using Piecolors to set the default net color does not update the colors on all pre-existing nets.*

Port — Select Port for Download Command (PC only)

Menu	Probe
Syntax	port lpt1 I lpt2 I lpt3 I user *user_command*
Abbreviations	none
See also	Download, Debugload, Defineprobe, Assignprobe

This command is exactly the same as the Port command found in the MakeBits program. It selects and initializes a port to be used by download. The options LPT1, LPT2, and LPT3 refer to the Xilinx-supplied parallel port download cable. The options COM1, COM2, COM3, and COM4 refer to the ports that the Xilinx-supplied XChecker serial cable uses. The User option permits you to download a bitstream to a destination other than the download cable. When the User option is selected, you are also prompted for a User command. This command string is executed when you start the Download command. The Download command produces a RBT (Rawbits) file for use with the user command. For example, a user command could be the following:

```
userprgm.exe -option %s
```

NOTE

DOS file names can be a maximum of eight characters.

In this example, userprgm.exe is a user program. Option is a command-line option of the program being started, and %s is replaced by a bitstream file written in RBT format. In this way, you can feed the RBT file into a custom program, which could then download it out the serial port to a PC card via the PC bus, and so forth. For more information concerning the Port command, consult "The MakeBits Program" chapter in Volume 2 of the *XACT Reference Guide.*

Table 3-17 PIE Display Items

Item	Default Color	Description
Reset	N/A	Restores default settings
Availablecolor	Green	The color of unused interconnect for the Show Available command
Routedusedcolor	Red	The color of unused stubs for the Show Routedused command
Fixedcolor	Green	The color of fixed interconnect, the non-programmable interconnect attached to block pins
Ppipcolor	Yellow	The color of the programmed PIPs
Bidicolor	Red	The color of the bi-directional buffers
Splittercolor	Red	The color of the longline splitter PIPs
Defaultnetcolor	Yellow	The color of nets created with the Addnet command
Uppipcolor	Green	The color of the unprogrammed PIPs
Unusedblockcolor	Green	The color of the unused blocks
Usedblockcolor	Yellow	The color of the used blocks
Matrixcolor	Red	The color of maxtrix highlighting used with the Show matrix command
PIPconnectionscolor	Green	The color of the PIP highlighting used with the Show PIP connections command
Magicboxcolor	Green	The color of the switch matrices (magic box)
Worldboxcolor	Red	The color of the World view box
Currentblockcolor	Red	The color of the configurable block being edited

Print — Create a Printable File of Display Information

Menu	Screen
Syntax	print *option filename*
Abbreviations	pr
See also	Draw, Printer, Printpic

This command creates a file that you can print from the operating system. Use the Binary option of the DOS Copy command to print PIC files on the PC. Four options determine the display information printed.

- Display prints the picture portion, LCA layout, or block configuration, of the screen.
- Screen prints all the information that appears on the screen, including menus and world map.
- World prints all of the LCA layout.
- Block prints the Block Editor display for a block.

If you enter Block, you must also enter the name of a block, for example, Print Block HB.

You must enter the file name from the keyboard. If the file already exists, Print replaces it with the new file and adds the file extension .pic.

The printer type, which determines the file format created, is set using the Printer command. If the printer type is not set before to using the Print command, the default printer type is used. Use the Printer command before the Print command to set the printer type.

It is possible that the default printer within XDE is not the one you have. The available printers are described in the printcap.xct file which is covered in the beginning of this chapter. Each printer described in this file shows up as an entry in the Printer menu. If you have a printer which does not appear on the menu for the Printer command, you may describe your printer to XDE with an entry in this file.

You can use the Print command to generate graphic printouts quickly. For example, the following syntax:

```
print display counter1
```

creates a printable file COUNTER1 from the display and saves it in the currently selected printer's format.

The following syntax:

```
printer okl92
print block dd demo
```

creates a printable file DEMO.pic from the Block Editor display of CLB DD and saves it in a format for the Okidata Microline 92 printer.

Printer — Set the Printer Type for Print Command

Menu	Profile
Syntax	printer *type*
Abbreviations	printe
See also	Print

The printer type determines the format used to create the printable file with the Print command. You can choose from one of the printer types that have built-in support in XDE release 5.0 listed in Table 3-18.

Refer to the "XACT Printer Support" section earlier in this chapter for a description of the printcap.xct file, which you can modify to support additional printers. You can change the initial value of the printer type in the xact.pro or editlca.pro file; the normal initial value is IBMgraph.

Table 3-18 Supported Printers

NEC	Nippon Electric Corp.
NECBIG	Nippon Electric Corp. (expanded)
MX80	Epson MX80
MX100	Epson MX100
FX80	Epson FX80
FX85	Epson MX85
FX86	Epson MX86
FX100	Epson FX100
FX185	Epson FX185
FX286	Epson FX286
RX80	Epson RX80
OKI92	Okidata Microline 92
OKI93	Okidata Microline 93
OKI292	Okidata Microline 292
OKI293	Okidata Microline 293
IBMgraph	IBM Graphics Printer
HPLASER	Hewlett–Packard Laserjet Printer
HPLPLUS	Hewlett–Packard Laserjet Plus Printer
HPLPPLUSLEGAL	HPLPLUS (legal size)
POSTSCRIPT	Postscript format

Queryblk — Request Data for Blocks

Menu	Blk
Syntax	queryblk *option blk*
Abbreviations	queryb, qb
See also	Cdata, Report

This command allows you several different ways to specify the blocks for which you are requesting information; by name, with the mouse, by using wildcards and with any of the eight tags associated with the Flagblk command.

The –long option displays detailed information about a block. You can use other options in conjunction with wildcards to specify a group of blocks based on a common characteristic. Information for a block is displayed only if it is one of the specified blocks and it matches all specified options. The options are listed in Table 3-19.

Table 3-19 Queryblk Command Options

Option	Report Characteristic
–All	Reports on all blocks in the design
–CLB	Reports on all CLBs in the design
–IOB	Reports on all IOBs in the design
–TBUF	Reports on all TBUFs in the design
–Used	Reports on all blocks in the design which have pins connected to them
–Unused	Reports on all blocks in the design that do have pins connected to them
–UsedFG	Reports on all CLBs that have at least one used function generator
–UnusedFG	Reports on all CLBs that have at least one unused function generator
–Configured	Reports on all blocks that are configured
–Unconfigured	Reports on all blocks that are unconfigured
–Locked	Reports on all blocks that are locked in place
–Unlocked	Reports on all blocks that are not locked in place
–Prohibited	Reports on all blocks that are prohibited
–Notprohibited	Reports on all blocks that are not prohibited
–Ainput	Reports on all blocks with an available input pin
–Aoutput	Reports on all blocks with an available output pin
–Net	Reports on all blocks attached to the selected net
–Synchronous	Reports on all blocks that have their function generators marked as synchronous
–IOB_ Enable_I_O	Reports on all IOBs which have been selected by the Flagblk IOB_Enable_O_I
IOB_Enable_T_I	Reports on all IOBs which have been selected using the Flagblk IOB_Enable_T_I
TBUF_ Disable_I_O	Reports on all TBUFs which have been selected using the Flagblk TBUF_Disable_I_O

Option	Report Characteristic
TBUF_ Disable_T_O	Reports on all TBUFs which have been selected using the Flagblk TUB_Disable_T_O
CLB_ Disable_SR_Q	Reports on all CLBs which have been selected using the Flagblk CLB_Disable_SR_Q
–Long	Displays Queryblk information in the long format
–Wide	Displays Queryblk information in the wide format; this is the default, only permitted if –long is not used

The wildcard (*) determines to which subset of blocks the options apply. The symbol * represents any characters. For example, typing G* as the block name represents all block names that begin with G. Figure 3-38 shows part of the result if you entered the following syntax:

```
queryblk -long -used aa ab ac b*
```

```
AB Unconfigured
     A= X=net2
     B= Y=
     C=
     D=
     K=
AC X: F Y: F:A:B G: Q: Set: RES: CLK:
 F= A+~B
     A=net1    X=net3
     B=net2    Y=
     C=
     D=
     K=
BC Unconfigured
     A= X=
     B= Y=net5
     C=
     D=
     K=
BD X:Q Y:g f:A:B:C: G:A:D Q:LATCH Set: RES: CLK:K
 F=A+B*C
 G=D@A
     A= X=net7
     B=net5    Y=net6
     C=
     D=
     K=
BE X:f Y:f:A;B:C:D Q: Set: RES: CLK:
 F=A+B+C+D
     A= X=
     B=net7    Y=net8
     C=net6
     D=
     K=
BF Unconfigured
     A=net8    X=
     B= Y=
     C=
     D=
     K=
```

Figure 3-38 Queryblk Example

Querygrid — Display the Grid Information of a Location

Menu	Screen
Syntax	querygrid *location*
Abbreviations	queryg, qg

You can specify locations with either the cursor or by name. Every cursor location, that is, name of a pin, PIP, or switching matrix pin, corresponds to a grid location that is based on X-Y coordinates of the LCA layout. This command displays the grid coordinates as well as the functional name of any cursor location. Table 3-20 shows examples of grid information.

Table 3-20 Querygrid Example

LOCATION	Querygrid Display		
	COORDINATES {X G Y}	NAME <pin>	NET CONNECTION (<netname>)
CLB pin	{79G58}	HD.C	(DATA)
IOB pin	{201G4}	PAD 5.1	(ADDRESS)
PIP	{193G78}	row.G.local.1:FC.X	(CLOCK)
empty space	{184G103}		()

Querynet — Display Information for a Net

Menu	Net
Syntax	querynet *option net*
Abbreviations	queryn, qn
See also	Report

You can specify nets by name, with the mouse, or with wildcards.

This command gives you several options for displaying the information for a specified net. Use the options in conjunction with a wildcard to select a group of nets based on a common characteristic. Information is displayed for a net only if it is one of the specified nets and it matches all specified options. The options are listed in Table 3-21.

Table 3-21 Querynet Command Options

Option	Characteristic
–All	Reports on all nets in the design
–Inputs	Reports on all nets that connect to IOB I pins
–Outputs	Reports on all nets that connect to IOB O pins
–Threestates	Reports on all nets that connect to IOB and/or TBUF T pins
–Unrouted	Reports on all nets with at least one unrouted load pin
–Nosource	Reports on all nets that have no source pin
–Nodest	Reports on all nets that have no destination pin
–Critical	Reports on all nets that were flagged as critical
–Tiechange	Reports on all nets that have had their delays changed with the Makebits –Tie command
–Tieadd	Reports on all nets that have been added with the Makebits –Tie command
–Baddelay	Reports on all nets with tildes (~)
–Locked	Reports on all nets that are locked
–Unlocked	Reports on all nets that are unlocked
–Delayless	Reports on all nets that have a delay less than a specified value
–Delaygreater	Reports on all nets that have a delay greater than a specified value
–Clocknets	Reports on all nets that connect to a clock pin, such as K, IK, or OK
–Probed	Reports on all nets that are connected to probes

The wildcard (*) determines to which subset of nets the options apply. The symbol * represents any characters. For example, typing A* as the net name represents all net names that begin with A.

Figure 3-39 shows sample results if you entered the following syntax:

```
querynet -all
```

```
M--- address...... TBUF.EC.1.O........ 3.3    DC.E
                   TBUF.EB.1.O
                   TBUF.ED.1.O

---- clock........ GCLK.O............. 2.7    FC.K
                                       2.7    FC.K
                                       2.7    FC.K

---- data0........ P20.I.............. 3.2    FC.B

---- data1........ P68.I..............11.3    GB.C
                                       7.6    HF.EC
                                      14.6    FH.A
                                       9.3    CG.B
                                      13.9    CA.RD

--C- dirreset .... HA.Y.............. 3.0    GD.C

-L-- loadless .... U DE.Y
S--- sourceless.......................***    EB.A
```

Figure 3-39 Querynet Example

The first four characters on a Querynet output line characterize various aspects of the net. Each character position is independent of the others. The four character positions and the possible values for each position are as follows.

Character 1 (leftmost)
S: Net has no source.
M: Net has multiple sources.

Character 2
L: Net has no loads.

Character 3
C: Net is marked critical (for MakeBits TIE, not APR).

Character 4
T: Net was modified by TIE in some way. Either pins were added to the net, or the delay to a pin already on the net was changed.

If the delay is specified as "***", it indicates that the pin is unrouted. If the delay is specified as "?", it means the net has been routed incorrectly. Use DRC to determine any routing errors in your design.

Querymargins — Report Delays for Paths Set by Setmargin

Menu	Timing
Syntax	querymargins
Abbreviations	querym, qm
See also	XDelay, Setmargins, Clearmargins, Savemargins

Querymargins reports all the path delays that have had a delay added to them by the Setmargins command (this delay is added to the calculated on-chip delays). The report contains the pad the path goes through, the clock net that is connected to the block at the start or end of the path, the margin (delay value) in nanoseconds added to that path, and the path type, either Pad-to-pad or Clock-to-pad. Below is a report example.

```
Block     Clock Net  Margin         Margin Type
  P28       CLOCK      10.5          Clocktopad
  P29       CLOCK      10.5          Clocktopad
  P30       CLOCK      10.5          Clocktopad
  P55       CLOCK       9.8          Padtosetup
```

See the XDelay Timing Analysis Program chapter in Volume 3 of the *XACT Reference Guide* for a description of the margins concept.

Queryprobe — Print a List of Probes and Information

Menu	Probe
Syntax	queryprobe
Abbreviations	querypr
See also	Defineprobe, Saveprobe, Readprobe

Queryprobe prints the current list of probes to the screen. It also prints the pin name, net name, and net delay, if applicable, associated with the probe(s). Below is an example.

```
 Name       Pin name     Net name      Delay (ns)

Channel_1     P48.0       ADDRESS0        9.7
Channel_2     P32.0       DATA2           2.5
Probe3       P15.0       (unassigned)
```

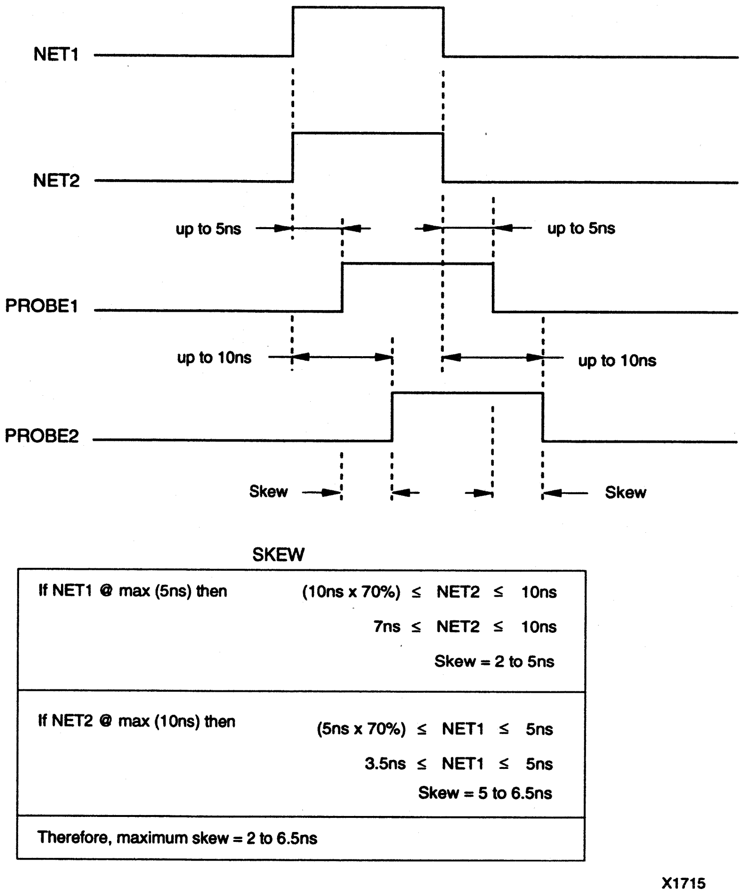
In this example, the third probe in the list, Probe3, has been defined but not assigned to a particular net.

NOTE

The delays reported by Queryprobe are worst case. The difference in delay of two probes seen externally may be different than simply the difference in their worst-case delays.

Figure 3-40 shows a probe timing example.

Name	Pin name	Net name	Delay (ns)
PROBE1	P21.0	NET1	5.0
PROBE2	P22.0	NET2	10.0



If NET1 @ max (5ns) then (10ns x 70%) ≤ NET2 ≤ 10ns

7ns ≤ NET2 ≤ 10ns

Skew = 2 to 5ns

If NET2 @ max (10ns) then (5ns x 70%) ≤ NET1 ≤ 5ns

3.5ns ≤ NET1 ≤ 5ns

Skew = 5 to 6.5ns

Therefore, maximum skew = 2 to 6.5ns

Figure 3-40 Probe Timing Example

Given the probe information in Figure 3-40, Probe1 and Probe2 are shifted to the left by 5 ns and 10 ns worst case, respectively. Originally, the signals Net1 and Net2 had no skew between themselves. After being probed, there can be from 2 ns to 6 ns of skew, depending on the delay tracking. Please refer to *The Programmable Logic Data Book* for more information about delay tracking.

To save the current probes to a file, use the Saveprobe command. To read a set of defined probes, use the Readprobe command.

Querysaveblk — Display Information about Saved Blocks

Menu	Blk
Syntax	querysaveblk *block_name*
Abbreviations	querys
See alsoq	Saveblk

This command displays information about all the blocks currently in temporary storage.

QueryTemplate — Display List of Current XDelay Options

Menu	Timing
Syntax	querytemplate
Abbreviations	qt
See also	Cleartemplate, Savetemplate, Readtemplate

This command prints a list of the currently defined XDelay options.

Readbits — Read the Specified Bitstream File

Menu	Probe
Syntax	readbits *file*
Abbreviations	readb
See also	Writebits

The Readbits command in the Probe menu reads the specified bitstream data file into the bitstream buffer. The XDE program supplies the extension .bit if it is not typed as part of the file name.

Since the start-up options are written into the bitstream, they are restored as the bitstream is read, even if they are not displayed in the options box.

Readcst — Read Constraint File Information (XC2000 and XC3000 only)

Menu	Aprcon
Syntax	readcst *file*
Abbreviations	readc
See also	Readscp, Writecst

This command reads an existing constraints file, .cst extension. It shows constraint information if Show Locked or Show Prohibited is on. By reading and writing CST files, you can apply and use constraints from the Design Editor to run APR.

Readmargins — Read Margin Delays from an MRG File

Menu	Timing
Syntax	readmargins
Abbreviations	readm
See also	Savemargins, Setmargins, Querymargins

This command reads the margin delays, which have been stored in a file with the Savemargins command, back into the LCA.

The MRG file stores the pad name, the clock net of the starting or ending block, the margin (delay) value, and the path type.

Readprobe — Read Set of Probes into the LCA File

Menu	Probe
Syntax	readprobe
Abbreviations	readpr
See also	Saveprobe, Defineprobe, Assignprobe

Readprobe reads a set of probes into the current design from a .prb file. The PRB file contains the probe name, the pin name associated with the probe, and the net name to which it has been assigned, if any. Readprobe does not save routing information. Therefore, the delays to various probes can change if you save the probes, load the design at a different time, and start the Readprobe command.

To save the current probes to a file, use the Saveprobe command. By not saving routing information, you can modify a design and still use a previous probe setup.

Readprofile — Set Editor Options to Settings in editlca.pro File

Menu	Profile
Syntax	readprofile
Abbreviations	readp
See also	Saveprofile, Settings

This command executes the commands in the file editlca.pro. This file contains commands for setting Editor options such as Autoroute, Mouse, Cursor, Printer, and Show. The editlca.pro file is normally created using the Saveprofile command.

Readscp — Read Schematic Constraint File Information (XC2000 and XC3000 only)

Menu	Aprcon
Syntax	readscp *file*
Abbreviations	reads
See also	Readcst, Writecst

This command reads schematic constraint file information.

ReadTemplate — Read XDelay Option File into Memory

Menu	Timing
Syntax	readtemplate *file*
Abbreviations	rt
See also	Querytemplate, Savetemplate, Cleartemplate

This command reads a list of XDelay settings from a template file. These settings are added to the template already present in memory. If the contents of the file are desired as the complete template, you should execute the Cleartemplate command first.

Redraw — Redraw the Display

Menu	Screen
Syntax	redraw
Abbreviations	red
See also	Hilight

This command redraws the display.

Report — Save Block or Net Information in a Text File

Menu	Misc
Syntax	report *filename command*
Abbreviations	rep
See also	Queryblk, Querynet, Querysavedblk, DRC, XDelay

Use Report to send Queryblk, Querynet, XDelay, Querysavedblk, and DRC information to a text file instead of displaying it on the screen. The *command* must be Queryblk, Querysavedblk, Querynet, DRC, or XDelay, followed by the required command parameters. For the full syntax of each command, see the Querysavedblk, Queryblk, Querynet, XDelay, and/or DRC command descriptions.

Restoreblk — Move Saved Information Back into a Block

Menu	Blk
Syntax	restoreblk *block_name*
Abbreviations	restoreb
See also	Saveblk

This command moves the contents of a predefined temporary storage space into a free block. XDE restores the block programming exactly as it was saved and connects the appropriate nets to the block. These nets are rerouted if Autoroute is on.

Route — Configure Interconnect for a Net

Menu	Net
Syntax	route *net*
Abbreviations	ro
See also	Autoroute, Routeblk, Routepin, Unroute, Editnet

This command configures the interconnect to connect all net pins unless any of the following situations occur.

- The net has no source pin.
- The net has more than one source pin.
- The net has no load pin.
- Available interconnects are inadequate to complete the necessary paths.

You can use Editnet to manually route nets that are not successfully routed by the automatic router.

Routeblk — Route All the Pins On a Block

Menu	Blk
Syntax	routeblk *block_name*
Abbreviations	routeb
See also	Route, Routepin, Unrouteblk

Routeblk routes all the pins on a specified set of blocks by using the Routepin command. You can move blocks in the Design Editor with Autoroute off and then automatically route the connections of certain blocks. This command accepts the wildcard character (*), so you can also route specific areas.

Routepin — Configure Interconnect for a Pin

Menu	Pin
Syntax	routepin *pin*
Abbreviations	routep, rp
See also	UnroutePin, Delpin, Autoroute, Route, Routeblk

This command configures the interconnect to connect a pin to the source pin of its net unless any of the following occurs.

- The pin net has no source pin.
- The pin net has more than one source pin.
- The pin net has no load pin.
- Available interconnect is inadequate to complete the necessary paths.

If the pin is the source of its net, the entire net is routed, and this command performs the same function as Route for a single pin or net.

Routepoint — Route from Point to Point

Menu	Pin
Syntax	routepoint *netplace*
Abbreviations	none

This command guides the XDE autorouter along a specified path. It permits the selection of points on a net and automatically routes between those points. You can then choose the points to route and include pins, PIPs, or magic boxes. Routepoint can tell the router to make a path from magic boxes A to B; it does not need to specify the pins on the magic boxes or intermediate PIPs that might need to be turned on.

Router — Change Usage Cost Associated with Routing Resources

Menu	Profile
Syntax	router *cost value*
Abbreviations	none

This command sets the costs that the XDE router uses to determine the best route for a net. Changing the settings of these costs can make the router behave in odd ways, but tuning them to obtain a desired behavior can be worthwhile. Table 3-22 lists the costs that you can change.

Table 3-22 Router Cost Settings

Cost	Default Value	Description
Reset	N/A	Resets all router costs to their initial values
Query	N/A	Displays the current settings for the router options
Fsadd	6	Non-Longline fixed cost segment
Eccadjust	8	Number subtracted from ECC (router estimated completion cost) if coming from a BIDI
Llcost	250	Longline fixed cost
Llpipcount	20	Minimum PIPs needed on a segment for it to be considered a Longline
Bidicost	0	BIDI buffer fixed cost
Firstbidi	2	Cost used to defer BIDIs from the source
MBlaneswitch	1	Incremental cost of a lane switch in a magic box

The cost that is most beneficial to adjust is the LLCOST variable; adjusting this down to the 50-100 range allows the router to use Longlines more freely.

Save — Save the Current Design State

Menu	Misc
Syntax	save *filename*
Abbreviations	none
See also	File, Quit

This command saves the current design state in the specified design file. Save prompts you before overwriting an existing file if you specify a file name other than the current file. The Editor automatically adds the LCA extension to the file name if you do not. If you do not specify a file, Save uses the current file name. After the design has been saved, Design Editor work can resume.

Since it is possible to leave MakeBits with a design in the tied state, File or Save issues a warning and prompts for a response if this is the case. This warning mitigates the possibility of overwriting an LCA file with a tied version of the design.

Saveblk — Save a Block to a Scratch Area

Menu	Blk
Syntax	saveblk *block_name*
Abbreviations	saveb
See also	Restoreblk, Querysaveblk

This command saves block functionality and connectivity in a temporary storage space. XDE then clears the block, disconnecting all nets connected to that block. An arbitrary number of storage spaces can exist with arbitrary names. Retrieve blocks using Restoreblk.

Savemargins — Save Margin Delays Set by Setmargins to an MRG File

Menu	Timing
Syntax	savemargins
Abbreviations	savem, sm
See also	Readmargins, Setmargins, Clearmargins

This command saves the margin delays that have been set previously with the Setmargins command to an MRG file.

Since these margins are not saved in the LCA file, they can be saved to an MRG file. When the design is loaded again, use the Readmargins command to read the margins into the design rather than re-entering them each time.

Savemargins stores the pad name, the clock net of the starting or ending block, the margin (delay) value, and the path type.

Saveprobe — Save Current Set of Probes into a PRB File

Menu	Probe
Syntax	saveprobe
Abbreviations	savepr
See also	Readprobe, Defineprobe, Assignprobe

Saveprobe saves the current set of probes into a PRB file. The PRB file contains the probe name, the pin name associated with probe, and the net name it has been assigned to, if any. It does not save routing information. Therefore, the delays to various probes might change if you save probes, load the design at a different time, and start the Readprobe command.

To read probes from a file, use the Readprobe command.

Saveprofile — Save Editor Option Settings in an editlca.pro File

Menu	Profile
Syntax	saveprofile
Abbreviations	savep
See also	Readprofile, Settings

This command saves your editor options in the editlca.pro file. The editlca.pro file is created in the current directory, if desired. The commands in this file set the Design Editor options and are normally executed automatically when you first use the Design Editor during an XDE session.

SaveTemplate — Save Defined XDelay Options to a File

Menu	Timing
Syntax	savetemplate *file*
Abbreviations	st
See also	Cleartemplate, Querytemplate, Readtemplate

This command saves the currently defined XDelay options, flagged nets, and other information to an XTM file. The file extension must be .xtm. After saving this information, you can retrieve it later with the ReadTemplate command.

Setmargins — Add Delay to Clock Paths that Start or End Off-Chip

Menu	Timing
Syntax	setmargins *path_type clock_net delay_margin*
Abbreviations	setm, sm
See also	XDelay, Querymargins, Clearmargins, Savemargins

Use this command to determine the delay of a path involving off-chip delays, when you know how much time it takes for the signal to propagate outside the LCA. You can add that delay value to the calculated on-chip delay and determine a total path-delay time.

With the Setmargins command, you can add a specified value to the delay of any clock path.

Setmargins prompts for the following information.

- Path type, which can be either Pad-to-setup or Clock-to-pad
- A clock net to be used as the reference clock, at which these delays begin or end
- Delay margin you want to add to the internal path-delay

See the XDelay Timing Analysis Program chapter in Volume 3 of the *XACT Reference Guide* for a complete description of the concept of margins.

Setmemory — Specify Power-On Values in ROM (XC4000 Only)

Menu	Config
Syntax	setmemory F I G I FG *hex_value*
Abbreviations	setme

Setmemory stores default values in the XC4000 CLB ROM. Once the device is configured, the ROM contains these preset values. If you do not enter a value, the specified memory (F, G, or FG) is cleared of its initial value. The F1 pin of the function generator is the MSB of the address.

The format of *hex_value* is:

x*value* or X*value*

where *value* is a string of one or more hexidecimal digits 0–9, a–f, or A–F, for example:

```
x0ff
XA
x68
```

Settings — Display Current Values of Editor Settings

Menu	Profile
Syntax	settings
Abbreviations	sett
See also	Saveprofile, Readprofile

This command displays the values of current Editor settings.

Show — Change the Display Options

Menu	Screen
Syntax	show *option*
Abbreviations	sh

This command includes options that affect the display. When you end the command, the display is updated according to the options you entered. If you enter Cancel, the display is not updated.

Pips Turns on the display of PIPs, switch matrices, and the fixed wire segments that connect to CLB and IOB pins.

NoPips Displays only blocks (CLBs, IOBs, and TBUFs) and routed nets.

World Turns on the world view display in the lower right corner of the screen.

NoWorld Displays the world view only when you use the mouse pan around the screen.

NeverWorld Prevents the world view from displaying, even when you pan around the screen.

Routed Displays only routed nets.

RoutedUsed Displays routed nets and all of the interconnect segments that they use.

Available Shows only the unused interconnect.

Bidis Highlights the bidirectional buffers. Use the Bidicolor option of the PieColors command to change the highlight color. This option has no effect unless the Pips option is also selected.

NoBidis Displays the bidirectional buffers in the same color as the PIPs.

Splitters Highlights the longline splitter PIPs. Use the Splitter color option of the PieColors command to change the highlight color. This option has no effect unless the Pips option is also selected.

NoSplitters Displays the longline splitters in the same color as the PIPs.

Matrix Highlights switch matrix pins when you select them with the cursor (in EditNet). Use the Matrixcolor option of the PieColor command to change the highlight color.

NoMatrix Disables the Matrix display.

Pipconnection Highlights Pips when you select them with the cursor (in EditNet)

NoPipconnection Prevents Pips from being highlighted.

Directional Draws PIPs as arrows, instead of as squares, to denote direction. ”>” represent PIPs that drive from vertical to horizontal (either right or left); ”^” represent PIPs that drive from horizontal to vertical (either up or down). Pips that remain square, are non-directional and can drive from any direction.

Nodirectional Draws PIPs as squares, regardless of directionality. The Pips option must also be selected for this option to have any effect.

Buffers Draws PIPs according to their buffering capability.

NoBuffers Disables the Buffers option.

Ratsnest Draws a line from the most recently programmed PIP to all net destinations. This option only has effect while Editnet is active.

NoRatsnest Turns off Ratsnest.

Locked Draws a thick border around any block that was locked in a constraint file for use with APR so you can see it more easily.

NoLocked Prevents Locked blocks from being shown.

Prohibited Draws a large X through the center of all prohibited blocks.

Noprohibited Does not show blocks which are the prohibited blocks.

XSmall Changes the viewing scale to extra-small magnification level.

Small Changes the viewing scale to small magnification level.

Medium Changes the viewing scale to medium magnification level.

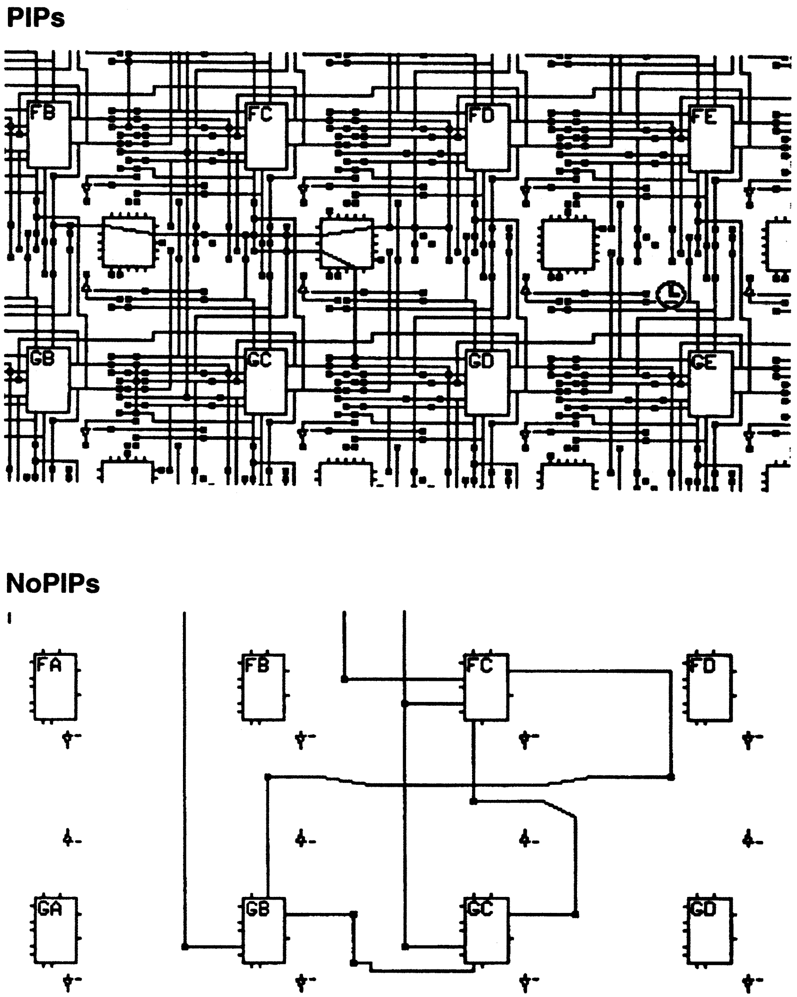
Large Changes the viewing scale to large magnification level.

XLarge Changes the viewing scale to extra-large magnification level.

In Zooms in one level of magnification.

Out Zooms out one level of magnification.

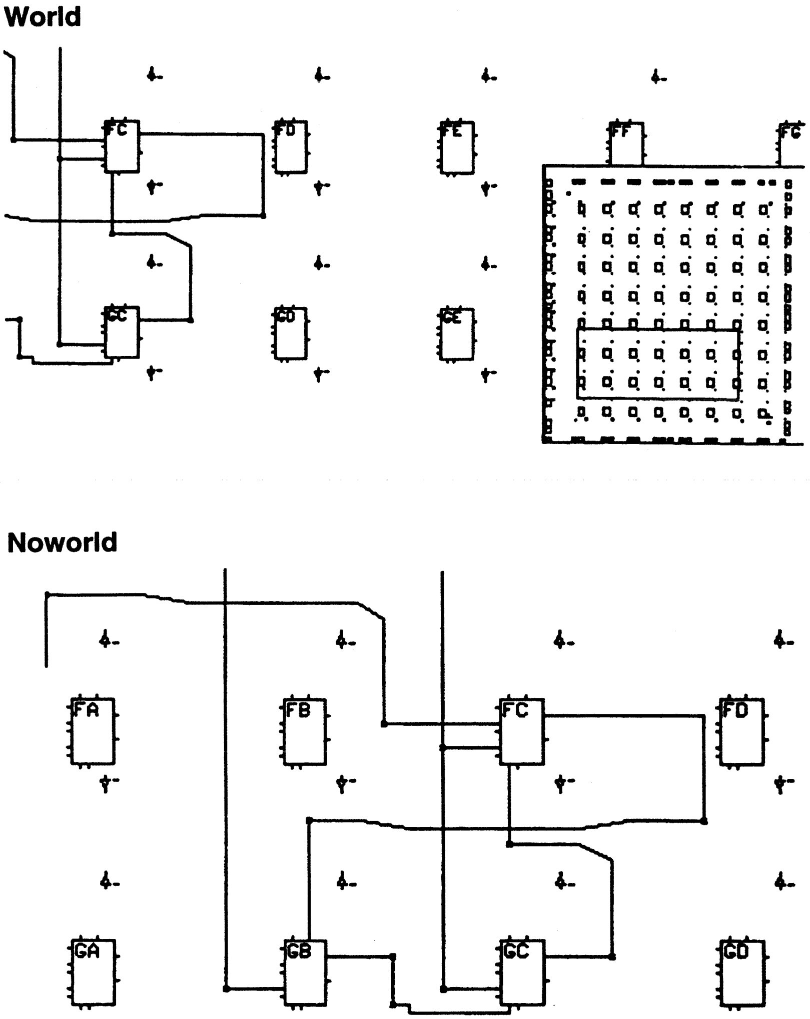
Figure 3-41, Figure 3-42, Figure 3-43, and Figure 3-44 illustrate some of these options.



PIPs

NoPIPs

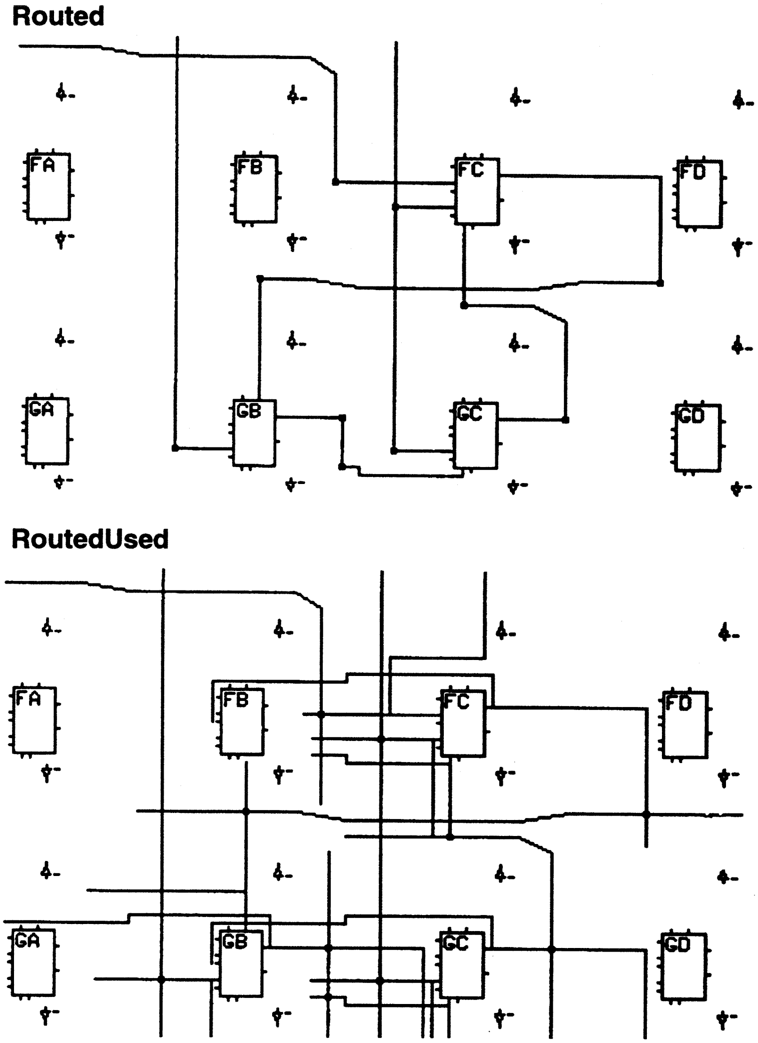
Figure 3-41 Pips vs NoPips



World

Noworld

Figure 3-42 World vs NoWorld


Routed

RoutedUsed

Figure 3-43 Routed vs RoutedUsed

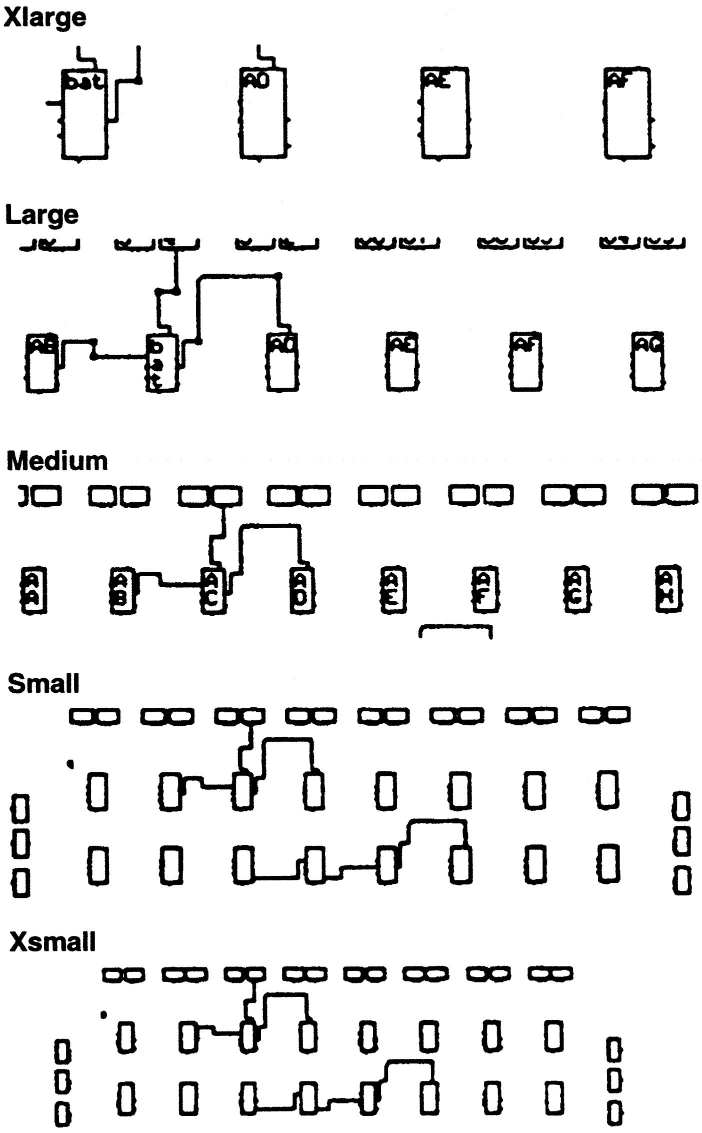


Figure 3-44 Magnification Options

ShowBlkConn — Highlight Block Connections in the World View

Menu	Blk
Syntax	showblkconn *color block*
Abbreviations	showb, shc

XDE highlights a block in a specific color, draws a line from the block pins to all connecting pins, and highlights all connected blocks.

NOTE *These connections are only displayed in the world view window.*

ShowNetConn — Highlight Nets in the World View

Menu	Net
Syntax	shownetconn *color net*
Abbreviations	shown, snc

Once you have selected a net, XDE highlights each block connected to that net in the specified color. Also, XDE draws a line from the net source to every load on the net.

NOTE *These connections are only displayed in the world view window.*

Snap — Make Pins or PIPs Snap to the Cursor

Menu	Screen
Syntax	snap Off | Pin | Pip | Pinorpip | Grid
Abbreviations	sn

When Snap is on and you release the mouse, the cursor stops moving and snaps to the closest item selected, depending on the snap setting selected. If Snap is turned off, the cursor stays in its last position when the mouse is released.

Speed — Select Device Speed Grade

Menu	Timing
Syntax	speed *speed_grade*
Abbreviations	spe

Speed sets the device speed grade for the current design. The speed grade is used by the XDelay and Autotime functions for determining signal propagation delays.

Splitnet — Split a Net into Two Distinct Nets

Menu	Net
Syntax	splitnet *sourcenet newnet pin*
Abbreviations	spl, sn

This command splits *sourcenet* into two distinct nets. You must enter the second net name. You can then select the pins, which currently exist on the sourcenet, that will move to the new net.

No source pin is created for the new net You are responsible for duplicating the logic that generated *sourcenet.*

StatusLine — Select Status Line Format

Menu	Screen
Syntax	statusline long I short
Abbreviations	status

The Long option provides all available status line information. The Short option limits the information that is displayed primarily to pin location. When you use the Short option, the mouse can track much faster, since the status line has to update less information.

With the Long option on, you can point at any section of routing to determine the name of the net. With the Short option on, you must point at the pins on that particular net to find out the name of the net.

Swapblk — Swap Configurations and Net Connections of Two Blocks

Menu	Blk
Syntax	swapblk *blk1 blk2*
Abbreviations	swapb, sb
See also	Copyblk, MoveBlk, SwapBlkGroup

The internal configuration and net connections of blk1 are moved to blk2, and vice versa. If either block has been named, the name is also moved. If the Autoroute option is on, new interconnects are configured accordingly. If any associated nets contain spurs, these spurs are deconfigured. The blocks must be the same type, either both CLBs or both IOBs.

Figure 3-45 shows the effect of Swapblk. In the top diagram, both blocks have been configured and renamed. The bottom diagram shows the effect of entering the following syntax.

```
swapblk dog cat
```

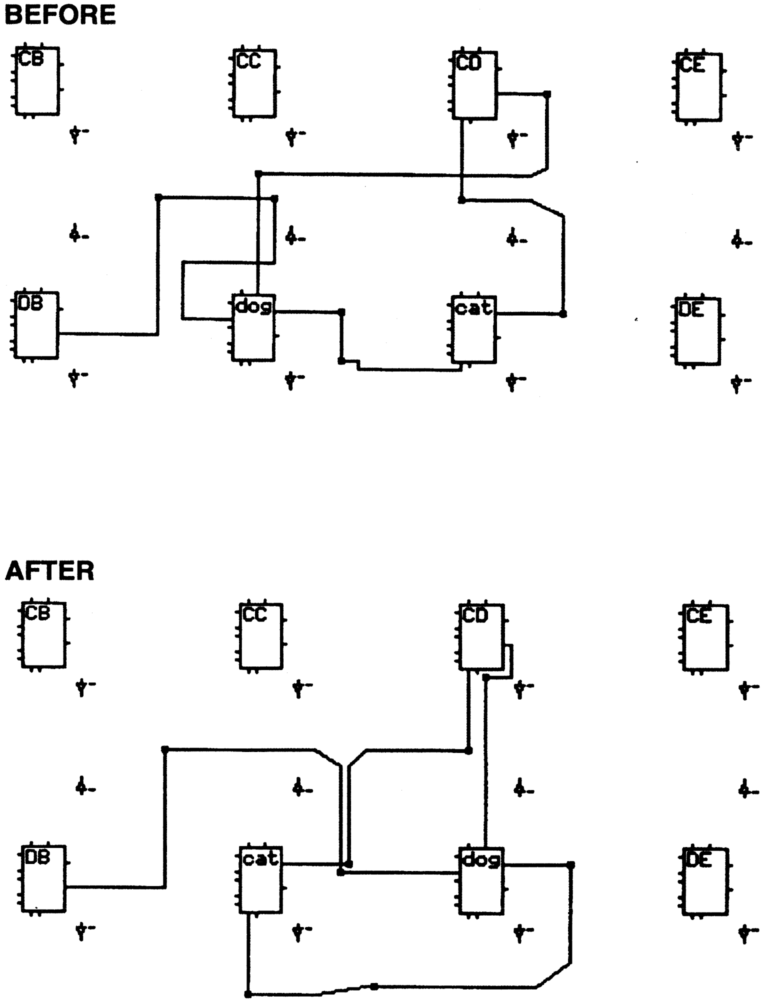


Figure 3-45 Swapblk Example

SwapBlkGroup — Swap Groups of Blocks

Menu	Blk
Syntax	swapblkgroup *group1_item group1_item2 ...* *group2_item1 group2_item2 blk1 blk2*
Abbreviations	none

SwapBlkGroup swaps groups of blocks. Rather than applying the SwapBlkGroup command individually to each pair of blocks, you can select all of the blocks in the first group, then all of the blocks in the second group. SwapBlkGroup swaps the first block from group one with the first block in group two, then the second block in group one with the second block in group two, and so on.

To swap CLBs in row A with CLBs in row D – that is, aa to da, ab to db, ac to dc, and so on – you could type:

```
swapblk aa da
swapblk ab db
swapblk ac dc
```

and so on. Alternatively, you could use the Swapblkgroup command.

```
swapblkgroup aa ab ac ... ↵
swap to block: da db dc ... ↵
```

Since the wildcard character (*) is supported with this command, swapping rows or columns of blocks becomes very easy. If you want to swap all the CLBs in column C with all the CLBs in column D, type the following.

```
swapblkgroup *c *d ↵
```

This example assumes that there are no other blocks names ending with the letter C or D.

SwapSig — Interchange the Functionality of Two Pins

Menu	Pin
Syntax	swapsig *pin1 pin2*
Abbreviations	swaps, ss
See also	Alignsig

The pins must be interchangeable inputs or outputs of the same CLB. The net locations of pin 1 and pin 2 are interchanged, and, if the Autoroute option is on, the interconnects are reconfigured appropriately. If either pin net contains a spur, the spur is deconfigured.

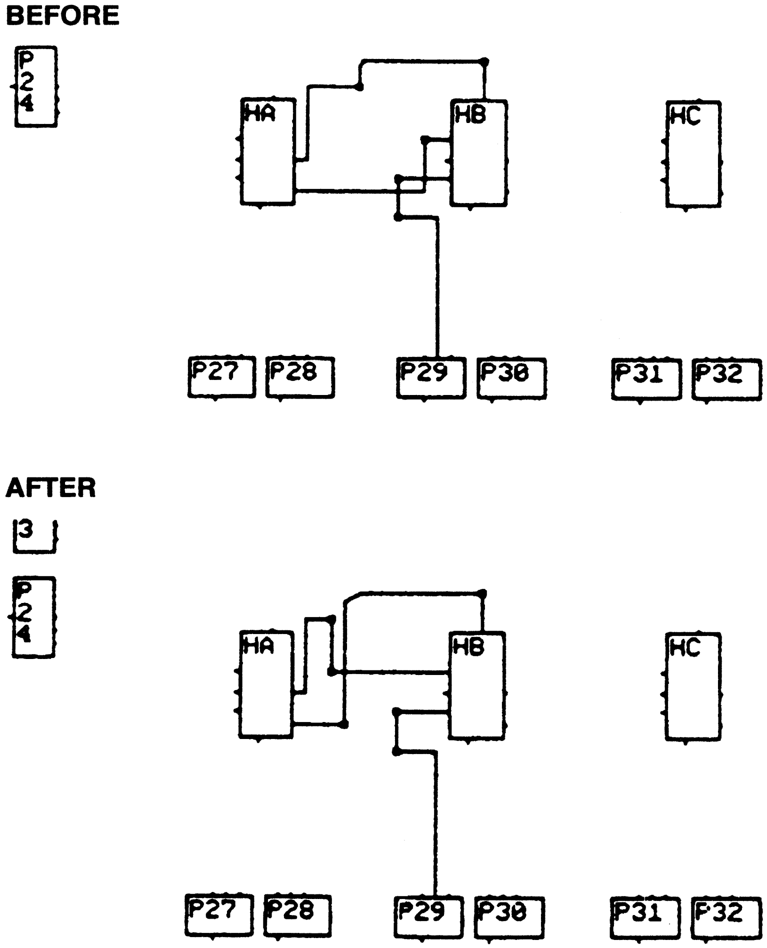
In addition, the pin CLB functions are interchanged. For example, if pin1 is the A input and pin2 is the B input, A and B are switched in the CLB configuration. If the configuration is such that the inputs cannot be swapped, the command is not performed.

Figure 3-46 shows the effect of Swapsig. The top diagram shows the initial interconnect and logic for block hb. The bottom diagram shows the results of entering the following.

```
swapsig hb.a hb.b
```

NOTE

The X and Y pins of an XC3000 series CLB can be swapped, but the internal connections to the multiplexers driving the X and Y pins of the CLB must also be swapped to accomplish this. Swapsig does this automatically.


BEFORE

AFTER

Figure 3-46 Swapsig Example

Switch — Toggle from PIE to Block Editor Display

Menu	Screen
Syntax	switch
Abbreviations	swi
See also	Mouse

If the PIE display is shown, Switch replaces it with the Block Editor display for the current block. If the Block Editor display is shown, Switch replaces it with the PIE display. If no current block is defined, nothing happens.

You can program a mouse button to perform the Switch function using the Mouse command.

UnAssignprobe — Unassign Probe from an Internal Net

Menu	Probe
Syntax	unassignprobe *probe*
Abbreviations	unassignpr, ua
See also	Assignprobe, Defineprobe, Undefineprobe

This command takes a pre-defined probe and disassociates it from the particular net to which it is assigned. XDE automatically unroutes the net associated with the probe; however, the probe remains defined.

Undefineprobe — Disassociate a Logical Name from an IOB

Menu	Probe
Syntax	undefineprobe *probe_name*
Abbreviations	undefinepr
See also	Defineprobe, Assignprobe, Saveprobe

This command disassociates a logical name from a particular IOB. The design editor removes this association from memory.

Undefineprobe brings up a list of valid probes currently defined. On the PC, use the arrow keys to move up and down the list. On the workstation, use the slider bar to move up and down the list.

Unhilight — Remove Highlighting from a Net Interconnect

Menu	Net
Syntax	unhilight *net*
Abbreviations	unhi
See also	Redraw, Hilight

This command redraws all of the net interconnects in the specified color and erases all highlighted metal stubs. This command is the inverse of Hilight.

Unroute — Deconfigure Interconnect for a Net

Menu	Net
Syntax	unroute *net*
Abbreviations	unr
See also	Autoroute, Delnet, Route

This command deconfigures the interconnect to all pins of the specified net. This command differs from Delnet in that the net still exists and the pins remain grouped. Unroute only deletes the routing interconnect and not the net or its pins.

UnRouteBlk — Unroute All the Pins on a Block

Menu	Blk
Syntax	unrouteblk *block*
Abbreviations	unrouteb
See also	Unroute, Routeblk

Unrouteblk unroutes all the pins on a specified set of blocks by using the Unroute pin command. This command accepts the wildcard character (*), so you can also unroute specific areas using this command.

UnroutePin — Deconfigure Interconnect for a Pin

Menu	Pin
Syntax	unroutepin *pin*
Abbreviations	unroutep, up
See also	Route, Routepin, Delpin

This command is the equivalent of Unroute for a single pin and the inverse command of Routepin. Unroutepin deconfigures the interconnect that connects the pin to the rest of the net interconnects. The pin must belong to a net. This command also deconfigures any of the net spurs.

UnShowBlkConn — Remove Block Connections from World View

Menu	Blk
Syntax	unshowblockconn *block*
Abbreviations	unshowb, usbc
See also	Showblkconn

This command removes the connections in the world view from the named block and its associated nets. This command accepts the wildcard character (*), so to clear all the Block connections from the world view, type the following.

```
unshowblkconn *
```

UnShowNetConn — Remove Net Connections from World View

Menu	Net
Syntax	unshownetconn *net*
Abbreviations	unshown, usnc
See also	Shownetconn

This command removes the connections in the world view from the named net or nets. This command accepts the wildcard character (*).

To clear all the net connections from the world view, type the following.

```
unshownetconn *
```

Updatehilight — Allow Nets to be Rehighlighted After Modification

Menu	Profile
Syntax	updatehilight on|off
Abbreviations	updatehl
See also	Hilight, Unhilight

If the Updatehilight variable is off, and a highlighted net is modified or routed differently, the old Hilight graphics remain or disappear, but no new highlighting takes place. If Updatehilight is on, the net is automatically re–highlighted when its definition or routing changes.

Weightnet — Assign a Routing Weightnet to a Net (XC2000 and XC3000 only)

Menu	Aprcon
Syntax	weightnet *weightnet net*
Abbreviations	weightnetn
See also	FlagIOB, Markblk, Marknet, Markpin

Weightnet assigns a weight to a net or set of nets. After you use the Writecst command, the assigned nets and their weights are stored in the CST file. APR reads this file to place the source and destinations of each net closer together, on the basis of weight and route the nets in order of highest weight. The default weight that APR gives a net is 3. The weight for a net flagged critical is 10. Weights can range from 1 to 99.

You should not weight many nets. Only weight the nets that are critical to the design performance. If too many nets are weighted, APR might place blocks too close together in an attempt to reduce delay and adversely affect both the overall placement and routability of the design. Also, if every net in a design is flagged critical, it is just as well not to flag any of the nets in the design. APR produces similar results in both cases, although the apparent score is higher in the first case.

Writebits — Save Current Configuration Bitstream

Menu	Probe
Syntax	writebits *file name*
Abbreviations	write
See also	Readbits

The Writebits command writes the contents of the bitstream buffer to the specified file. The file extension must be .bit. The XDE program automatically appends the .bit extension if it is not supplied.

Writecst — Write Constraint Information to a File (XC2000 and XC3000 only)

Menu	Aprcon
Syntax	writecst *file*
Abbreviations	writec
See also	Readcst, Readscp

This command writes all constraint information to a file. See The Automatic Place and Route Program chapter in Volume 2 of the *XACT Reference Guide* for more information about APR and constraint files.

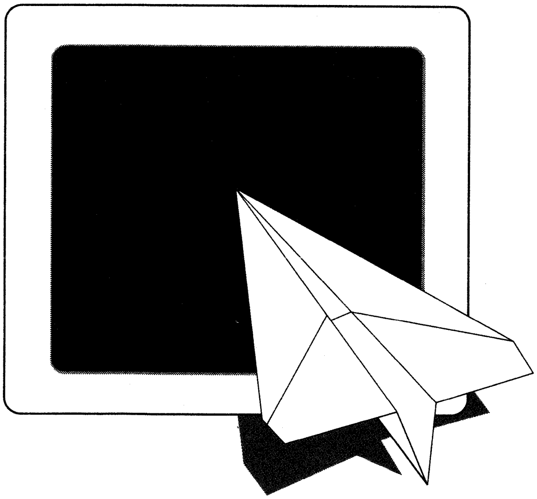
XDelay — Report Detailed Timing Information of LCA Design

Menu	Timing
Syntax	xdelay *option*
Abbreviations	none
See also	Report, Setmargin, Readmargin

With the XDelay command, you can obtain detailed timing information about your design. The report contains the worst-case path delays for the set of chosen paths. You can choose paths by selecting individual starting points for paths, or by specifying a certain type of path type to follow. For example, trace every input pad that eventually connects to a flip-flop, or from all flip-flop outputs that connect to flip-flop inputs.

You can also use XDelay to analyze a design's overall performance. The Analyze option calculates maximum clock frequencies for all the clocks in the design. In addition, it lists the worst-case path delays and the number of levels of logic through them.

After selecting the XDelay command, a menu of options, described in the XDelay chapter of this manual appears. Select the desired options or enter them at the command line, and then select Done at the top of the screen to execute XDelay.



XACT Reference Guide, Volume 3

Index

Index