Lattice™
Semiconductor
Corporation

Data I/O and pDS+ Design and Simulation Environment

User Manual

Version 3.0

Technical Support Line: 1-800-LATTICE or (408) 428-6414
pDS2102-UM Rev 3.00

Lattice Semiconductor Corporation
5555 N.E. Moore Court
Hillsboro, OR 97124
(503) 681-0118

April 1996

Limited Warranty

Lattice Semiconductor Corporation warrants the original purchaser that the LSC software shall be free from defects in material and workmanship for a period of ninety days from the date of purchase. If a defect covered by this limited warranty occurs during this 90-day warranty period, Lattice Semiconductor will repair or replace the component part at its option free of charge.

This limited warranty does not apply if the defects have been caused by negligence, accident, unreasonable or unintended use, modification, or any causes not related to defective materials or workmanship.

To receive service during the 90-day warranty period, contact Lattice Semiconductor Corporation at:

Phone: 1-800-LATTICE
Fax: 1 (408) 944-8450
E-mail: applications@latticesemi.com

If the Lattice Semiconductor support personnel are unable to solve your problem over the phone, we will provide you with instructions on returning your defective software to us. The cost of returning the software to the Lattice Semiconductor Service Center shall be paid by the purchaser.

Limitations on Warranty

Any applicable implied warranties, including warranties of merchantability and fitness for a particular purpose, are hereby limited to ninety days from the date of purchase and are subject to the conditions set forth herein. In no event shall Lattice Semiconductor Corporation be liable for consequential or incidental damages resulting from the breach of any expressed or implied warranties.

The Lattice Semiconductor pDS+ Synario/ABEL software for the Synario/ABEL design environments is designed to allow the user to transfer designs from the Synario/ABEL design environment to Lattice Semiconductor ispLSI and pLSI devices. This Lattice Semiconductor software cannot be used independently of the Synario/ABEL software.

Purchaser’s sole remedy for any cause whatsoever, regardless of the form of action, shall be limited to the price paid to Lattice Semiconductor for the Lattice Semiconductor pDS+ Synario/ABEL software.

The provisions of this limited warranty are valid in the United States only. Some states do not allow limitations on how long an implied warranty lasts, or exclusion of consequential or incidental damages, so the above limitation or exclusion may not apply to you.

This warranty provides you with specific legal rights. You may have other rights which vary from state to state.

Table of Contents

Preface 7

What Is in this User Manual 8

Where to Look for Information 8

User Manual Conventions 9

Related Documentation 10

Lattice Semiconductor 10

Data I/O 10

Synario 10

ABEL 10

Synario and ABEL 10

Chapter 1 Introduction 11

pDS+ Fitter and Synario Design Flow 12

pDS+ Fitter and Synario Design Flow Steps 13

pDS+ Fitter and ABEL-WIN Design Flow 14

pDS+ Fitter and ABEL-WIN Design Flow Steps 15

pDS+ Fitter and ABEL-DOS/ABEL-SUN Design Flow 16

pDS+ Fitter and ABEL-DOS/ABEL-SUN Design Flow Steps 17

pDS+ Synario/ABEL Files and Processes 18

Compilation Process Files 18

Simulation Files 18

Synario 18

ABEL 18

Device Programming File 18

Design Process Manager (DPM) 18

Parameter File 19

Partitioner 20

Router 20

Test Vectors 21

Synario 21

ABEL 21

Post-Route Simulation Netlist 22

Device Programming 23

Chapter 2 pDS+ Fitter and Synario/ABEL Environment Setup 24
System Requirements for pDS+ Synario/ABEL. 24
PC Software Requirements 24
PC Hardware Requirements. 24
UNIX Workstation Software Requirements 25
UNIX Workstation Hardware Requirements. 25
Installing pDS+ Synario 26
Synario Device Kit 26
Environment Setup. 26
Additional Requirements 26
Installing pDS+ ABEL-WIN 27
ABEL Device Kit. 27
Environment Setup. 27
Installing pDS+ ABEL-DOS 28
ABEL Memory Requirement. 28
Environment Setup. 28
Installing pDS+ ABEL-SUN 29
ABEL Memory Requirement. 29
Environment Setup. 29
Flex License Environment Configuration 29
Software Update Service 30
Customer Hotline 30

Chapter 3 Design Entry 31
Design Attribute Overview 32
Design Attributes in Synario Schematic Designs 33
Design Attributes in ABEL-HDL 34
Net Attributes 34
Path Attributes. 34
Symbol Attributes 35
Pin Attributes. 35
Synario/ABEL-WIN Design Attribute Restrictions
in Hierarchical Designs. 36
ABEL-HDL Compatibility 39
ABEL–HDL Dot Extensions and Mapping 39
ABEL Signal Attributes 43
ABEL-HDL Exceptions 43

Chapter 4 Design Compilation 44
Compiler Control Options Overview. 45
ABEL-HDL Syntax Examples 47
Synario and the pDS+ Fitter 50
Specifying Compiler Control Options 50
Invoking the pDS+ Fitter. 51
Controlling the pDS+ Fitter 51
Specifying the Parameter File. 51
Synario Design Example 52

ABEL-WIN and the pDS+ Fitter 54
Invoking the pDS+ Fitter 55
Controlling the pDS+ Fitter 55
Specifying the Parameter File 55
ABEL-WIN Design Example 56
ABEL-DOS and the pDS+ Fitter 58
Supported SmartPart Options 58
Invoking the pDS+ Fitter 59
Controlling the pDS+ Fitter 60
Specifying the Parameter File 60
ABEL Design Example 60
ABEL-SUN and the pDS+ Fitter 63
Supported SmartPart Options 63
Invoking the pDS+ Fitter 64
ABEL-SUN 4.3 64
ABEL-SUN 5.1 65
Controlling the pDS+ Fitter 66
ABEL-SUN 4.3 66
ABEL-SUN 5.1 66
Specifying the Parameter File 66
ABEL-SUN Design Example 66

Chapter 5 Simulation 67
Functional Simulation. 68
Synario Designs 68
ABEL-WIN Designs 69
ABEL-DOS/ABEL-SUN Designs 69
Timing Simulation 70
Synario Designs 70
ABEL-WIN, ABEL-DOS, and ABEL-SUN Designs 70
Lattice Semiconductor-Specific Simulation Signals 71
!XRESET. 71
XTEST_OE 71

Preface

This manual describes how to use pDS+™ Synario/ABEL software to create designs for the Lattice Semiconductor Corporation (LSC) family of ispLSI® and pLSI® devices. This manual describes how to apply LSC Design Attributes and Compiler Control Options to create designs to download into LSC programmable logic devices (PLDs).

This manual is intended for use by design engineers who are knowledgeable in system design and architecture. It assumes that you are familiar with the Data I/O Synario™ software, the Data I/O ABEL™ software, the Lattice Semiconductor pDS+ Fitter (also referred to as the Compiler), and LSC device architecture. If you are not knowledgeable in these design methodologies, refer to the appropriate reference material listed under “Related Documentation.”

What Is in this User Manual

This user manual contains information on the following topics:

- Design Flows for Synario/ABEL Designs to the pDS+ Fitter
- Environment Setup
- Applying LSC Design Attributes and Compiler Control Options
- Design Compilation
- Design Examples

Where to Look for Information

Chapter 1, Introduction – Provides an overview of the design flow using Synario or ABEL and the LSC pDS+ Fitter.

Chapter 2, Data I/O and pDS+ Environment Setup – Explains how to set up the correct environment for running the pDS+ Fitter on designs created in Synario or ABEL.

Chapter 3, Design Entry – Provides an ABEL-HDL syntax example for the LSC Design Attributes and explains how to apply these attributes in designs created using Synario or ABEL. It also describes compatibility between Data I/O menus and LSC Design Attributes.

Chapter 4, Design Compilation – Provides an ABEL-HDL syntax example for the LSC Compiler Control Options. It explains how to apply Compiler Control Options in designs created using Synario or ABEL and contains step-by-step design examples. It also describes how to invoke the pDS+ Fitter.

Chapter 5, Simulation – Explains the simulation process for designs that target LSC ispLSI and pLSI devices.

User Manual Conventions

The conventions in this user manual are defined in the table below.

Convention	Definition and Usage
Italics	Italicized text represents variable input. For example: *file_name*.tf You must replace *file_name* with the file name you used in your design. The beginning of procedures and book titles also appear in italics. For example: *pDS+ Fitter User Manual*
Bold	Valuable information may be boldfaced for emphasis. Commands are always shown in boldface. For example: **Add ➠ Symbol Attribute**
`Courier Font`	Monospaced (Courier) font indicates file and directory names and text that the system displays. This font is also used in syntax examples. For example: `plsi property ’clk Clk clk0’;`
`Bold Courier`	Bold Courier indicates text you type in response to system prompts. For example: **`cp /`*`<pdsplus_install_path>`*`/abel/* ...`**
|...|	Vertical bars indicate options that are mutually exclusive; you can select only one. For example: REGTYPE GLB|IOC
“Quotes”	Titles of chapters, or sections in chapters, in this user manual are shown in quotation marks. For example: See Chapter 3, “Design Entry.”
✍ ***NOTE***	Indicates a special note.
▲ **CAUTION**	**Indicates a situation that could cause loss of data or other problems.**
❖ **TIP**	Indicates a special hint that makes using the software easier.

Related Documentation

In addition to this user manual, the following documentation is helpful when using pDS+ Synario/ABEL:

Lattice Semiconductor

- *ISP Daisy Chain Download Reference Manual*
- *Lattice Data Book*
- *pDS+ Fitter Installation Guide – (PC or UNIX)*
- *pDS+ Fitter User Manual*

Data I/O

Synario

- *ABEL-HDL Reference Synario Universal FPGA Design System*
- *Getting Started Synario Universal FPGA Design System*
- *Project Navigator User Manual Synario Universal FPGA Design System*
- *Schematic Editor Reference Synario Universal FPGA Design System*
- *Schematic Entry Reference Synario Capture System and ECS*
- *Schematic Entry User Manual Synario Capture System and ECS*
- *Synario User Manual*
- *Verilog HDL Reference Synario Universal FPGA System*
- *Verilog Simulator User Manual Synario Universal FPGA Design System*

ABEL

- *ABEL-HDL Reference*
- *Getting Started ABEL Design Software*
- *Project Navigator User Manual ABEL Design Software*

Synario and ABEL

- *Equation and JEDEC Simulators User Manual Synario and ABEL*
- *PLD Device Kit Manual Synario and ABEL*
- *pLSI Device Kit Manual*
- *Waveform Tools Manual Synario and ABEL*

Chapter 1 *Introduction*

The pDS+™ Synario/ABEL solution provides an interface between the Data I/O Synario/ABEL design software, and the pDS+ Fitter to generate programming files for the Lattice Semiconductor Corporation (LSC) ispLSI® and pLSI® families of high-performance, high-density programmable logic devices (PLDs).

This chapter provides an overview of the LSC functions and design flow using the pDS+ Fitter. It contains information on the following topics:

- pDS+ Fitter and Synario Design Flow
- pDS+ Fitter and ABEL-WIN Design Flow
- pDS+ Fitter and ABEL-DOS/ABEL-SUN Design Flow
- pDS+ Synario/ABEL Files and Processes

pDS+ Fitter and Synario Design Flow

Figure 1-1 illustrates a typical design flow from design entry through device programming using pDS+ Synario.

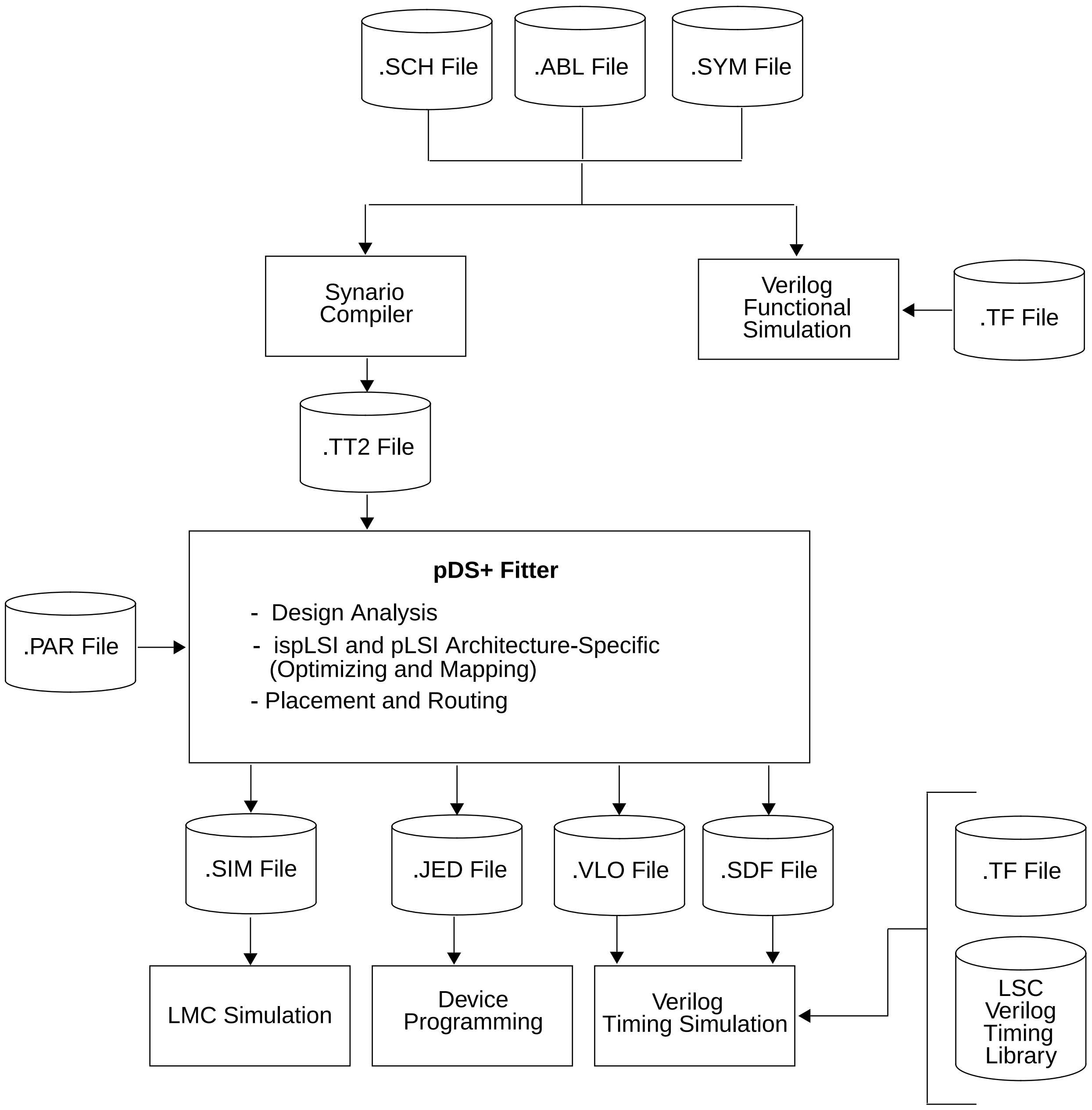

Figure 1-1. Synario to the LSC pDS+ Fitter Design Flow

pDS+ Fitter and Synario Design Flow Steps

The LSC pDS+ Fitter is used in conjunction with Synario designs targeting LSC ispLSI and pLSI devices. Figure 1-1 displays the design flow from Synario to the LSC pDS+ Fitter. The design flow steps are as follows:

1. Create a design.
2. Create a *design_name*.tf file.
3. Perform functional simulation (optional).
4. Select an LSC device.
5. Add the desired LSC Design Attributes.
6. Select **Fit Design** to generate the *design_name*.tt2 file, to add the desired Compiler Control Options, and to invoke the LSC pDS+ Fitter.
7. Perform timing simulation (optional).
8. Program an LSC device.

pDS+ Fitter and ABEL-WIN Design Flow

Figure 1-2 illustrates a typical design flow from design entry through device programming using pDS+ ABEL.

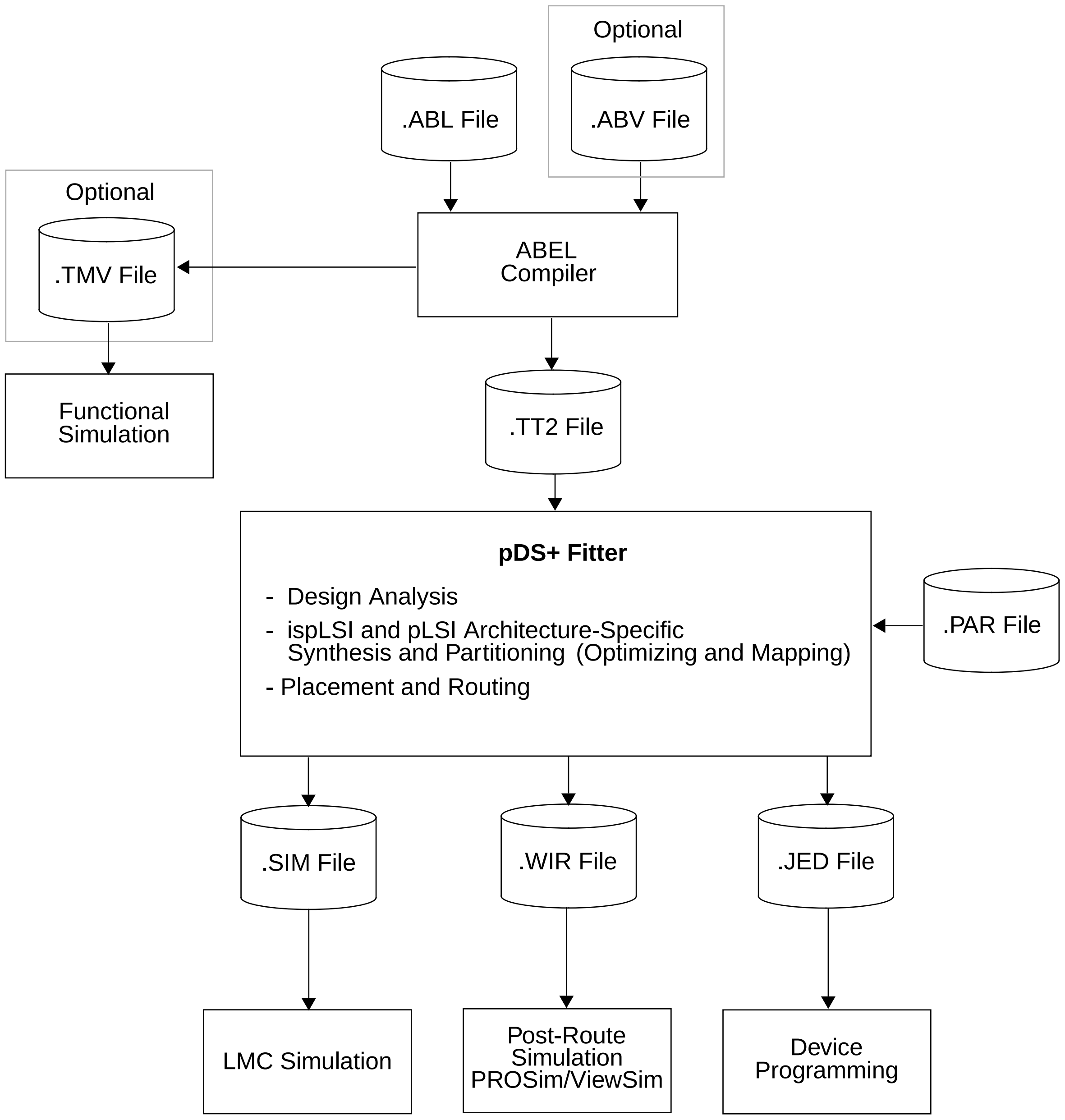

Figure 1-2. Design Flow ABEL-WIN to pDS+ Fitter

pDS+ Fitter and ABEL-WIN Design Flow Steps

The LSC pDS+ Fitter is used in conjunction with ABEL-WIN designs targeting LSC ispLSI and pLSI devices. Figure 1-2 displays the design flow from ABEL-WIN to the pDS+ Fitter. The design flow steps are as follows.

1. Create an ABEL-HDL design.
2. Create test vectors. You can either include test vectors in your ABEL-HDL design, or in the *file_name*.abv file. The ABEL compiler generates the *file_name*.tmv as the input file for functional simulation.
3. Perform functional simulation (optional).
4. Select an LSC device.
5. Add the desired LSC Design Attributes.
6. Select **Fit Design** to generate the *file_name*.tt2 file, to add the desired Compiler Control Options, and to invoke the LSC pDS+ Fitter.
7. Perform timing simulation (optional).
8. Program an LSC device.

pDS+ Fitter and ABEL-DOS/ABEL-SUN Design Flow

Figure 1-3 illustrates a typical design flow from design entry through device programming using pDS+ ABEL.

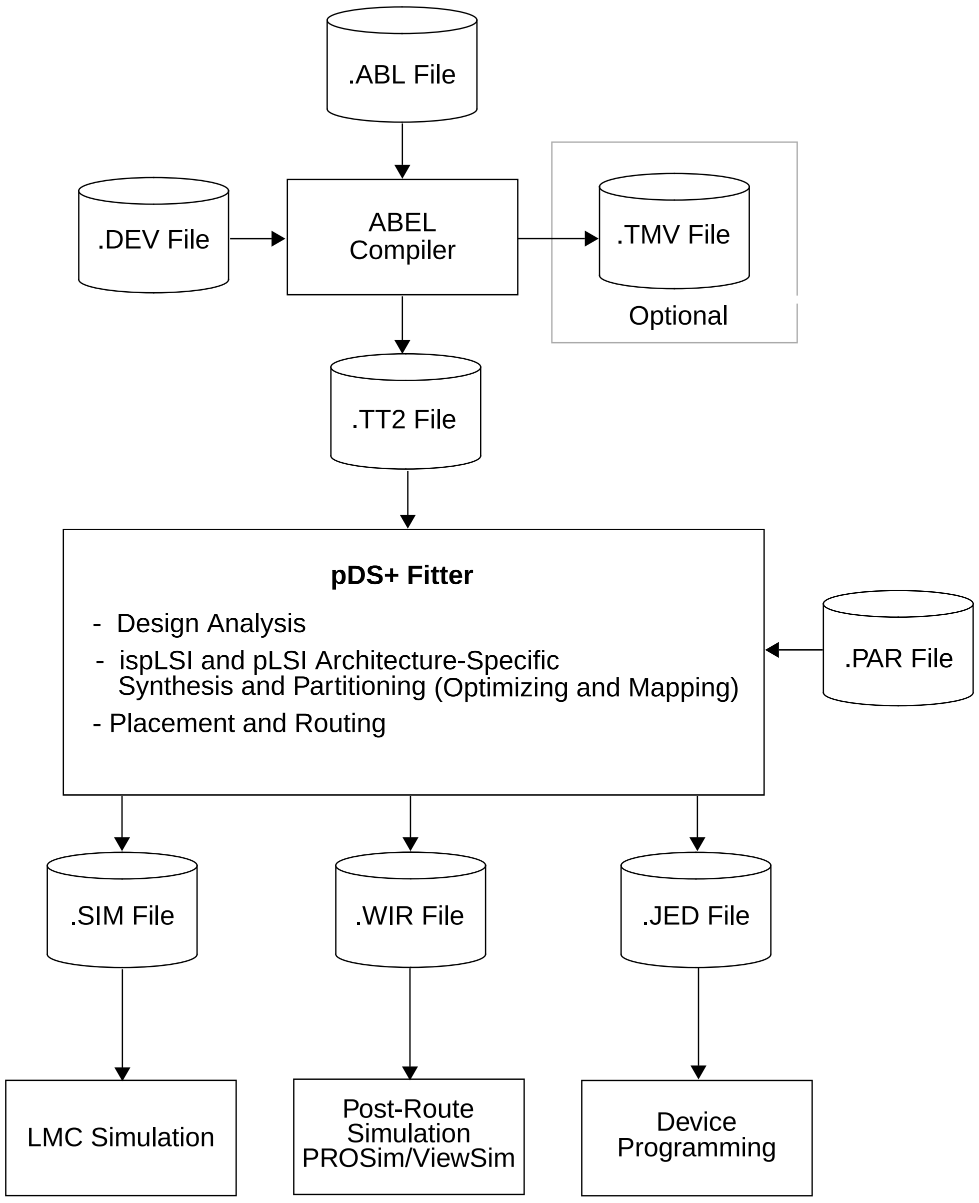

Figure 1-3. Design Flow ABEL-DOS/ABEL-SUN to pDS+ Fitter

pDS+ Fitter and ABEL-DOS/ABEL-SUN Design Flow Steps

The LSC pDS+ Fitter is used in conjunction with ABEL-DOS/ABEL-SUN designs targeting LSC ispLSI and pLSI devices. Figure 1-3 displays the design flow from ABEL-DOS/ABEL-SUN to the pDS+ Fitter. The design flow steps are as follows.

1. Enter your design description in ABEL-HDL format.
2. Add the desired LSC Design Attributes and Compiler Control Options as ABEL properties in the ABEL-HDL source file.
3. Specify an LSC ispLSI or pLSI device in the ABEL-HDL source file or from the ABEL menu.
4. Compile ABEL-HDL into a *file_name*.tt2 file using the ABEL compiler.
5. Invoke the LSC pDS+ Fitter.
6. Perform timing simulation (optional).
7. Program an LSC device.

pDS+ Synario/ABEL Files and Processes

Compilation Process Files

- ***file_name*.tt2** – The .tt2 file is a required input file when running the LSC pDS+ Fitter. The .tt2 file contains your Design Attributes.
- ***file_name*.par** – The Parameter File (.par) is an optional input file for one or more sets of Compiler Control Options. The .par file contains the design control options that can be used to improve design routability and device resource utilization.

Simulation Files

Synario

- ***file_name*.tf** – The .tf file is a user-created test fixture input file used for Verilog functional and timing simulation.
- ***file_name*.vlo** – The .vlo file is output by the pDS+ Fitter and is a required input file for Verilog timing simulation.
- ***file_name*.sdf** – The .sdf file is output by the pDS+ Fitter and is a required input file for Verilog timing simulation.

ABEL

- ***file_name*.abv** – The .abv file is a user-created test vector input file. This file is only available within ABEL-WIN. Use the .abv file if you do not wish to include test vectors in your *design_name*.abl file.
- ***file_name*.tmv** – The .tmv file is a required input file for performing simulation on ABEL designs. This file is generated by the ABEL compiler if you are using ABEL-WIN, ABEL-DOS, or ABEL-SUN.
- ***file_name*.1** – The .1 file in the WIR directory is a required input file for Viewlogic timing simulation on ABEL designs.

Device Programming File

- ***file_name*.jed** – The .jed file is generated by the pDS+ Fitter after placing-and-routing your design. You can download the JEDEC (.jed) file directly into an LSC ispLSI or pLSI device.

Design Process Manager (DPM)

The Design Process Manager (DPM) (the pDS+ Fitter) guides the compilation process. When the fitter is accessed through the Synario/ABEL Fit Process Properties Editor menu (in Synario or ABEL-WIN), or the ABEL Fit menu (in ABEL-DOS/ABEL-SUN), all relevant files and parameters are passed to appropriate compiler modules. Once a design routes successfully, the fitter merges various report files into one final report file titled *file_name*.rpt.

The Compiler Control Options direct various compiler processes. The Design Attributes provide additional control and determine design parameters such as pin designations and critical path specifications. The fitter handles the following:

- Parameter File
- Partitioner
- Router
- Fusemap Generation
- Test Vector Merging
- Post-Route Simulation Netlist File Generation

Parameter File

The optional Parameter File *(file_name.*par) contains one or more sets of Compiler Control Options that are used as alternatives to the properties specified during design entry. You can use this file to automatically run different iterations of your design for better optimization. The Parameter File overrides the Compiler Control Options specified in your source file and assumes default values for any unspecified properties. See the *pDS+ Fitter User Manual* for information on how to use Compiler Control Options in the Parameter File.

Partitioner

The partitioner performs the following functions:

- Optimizes logic equations by:
  - Minimizing multi-level logic
  - Extracting XOR logic to take advantage of physical XOR gates in the device
  - Using XORs to reduce logic through function inversion (where possible)
  - Mapping parallel registers into a single register
  - Eliminating unused registers and outputs
- Clusters the partitioned logic according to common clocks, output enable (OE) signals, reset signals, and fixed pin constraints
- Groups logic to fit within Generic Logic Blocks (GLBs) and IOCs
- Generates a netlist for the router

Router

Once the design is partitioned, the router reads the partitioner-generated file which contains design netlist information, placement constraints, and routing constraints. The router then performs the following functions:

- Assigns I/O pins to specific IOC locations
- Assigns logic to specific GLB locations
- Interconnects GLBs and IOCs
- Generates a place-and-route output netlist

After routing, the compiler performs a post-route mapping and prepares the files for input to the fusemap generation program.

Test Vectors

Synario

Use the Synario test fixture file, *file_name*.tf for both functional and timing simulation. Create these test fixtures by using either the Synario text editor, or an external text editor. (If you use an external text editor, you must import this into your Project Directory.) For further information on the *file_name*.tf file and simulating Synario designs targeting Lattice Semiconductor ispLSI and pLSI devices, see Chapter 5, “Simulation” and the *Synario User Manual*.

ABEL

Use the ABEL test vector file, *file_name*.tmv for functional verification. The *file_name*.tmv file is automatically created when you specify test vectors in your ABEL source file.

Converting the test vectors from the *file_name*.tmv file to a JEDEC file is the final step of the device fitting process. If there are no test vectors included in the ABEL source file, the *file_name*.tmv file is not created and no test vector section will be included in the JEDEC file. If the *file_name*.tmv file is present, the JEDEC file consists of a fusemap section and a test vector section. The combined information can be used to program and verify the functionality of the device. See Chapter 5, “Simulation,” for further information on performing timing simulation on ABEL designs using pDS+ ABEL.

Post-Route Simulation Netlist

The pDS+ Fitter can generate output netlists for post-route timing simulation in the following formats: EDIF, LDF, OrCAD, PREROUTE_LDF, SIM, VERILOG, VHDL, and VIEWLOGIC. Use the Compiler Control Option, OUTPUT_FORM to specify the type of output netlist format to be generated. You can specify several output netlist formats by typing them on the same line in the Parameter File. For example:

```
OUTPUT_FORM EDIF SIM
```

See Chapter 4, “Design Compilation,” and the *pDS+ Fitter User Manual* for further information. The following list gives a description of what each variable does:

- EDIF – Generates an EDIF 2 0 0 format netlist (*design*.edo) file.
- LDF – Generates a *design*.ldf file, representing the post-route design if the design routes, and pre-route design if the design fails to route. The LSC Design Format (LDF) file can be exported to the pDS software package. The pDS package allows you to manually partition the logic in a device to possibly improve routability. This is useful if a design requires significant manual intervention.
- ORCAD – Generates an OrCAD INF format netlist (*design*.ifo) file which can be used for post-route simulation with OrCAD’s VST simulator.
- PREROUTE_LDF – Generates a *design*.ldf file, representing the pre-route partitioned logic design.
- SIM – Generates a *design*.sim file to use for board-level simulation with Logic Modeling Corporation (LMC) models.
- VERILOG – Generates an SDF format netlist (*design*.sdf) file as well as a Verilog format netlist (*design*.vlo) file for use with any Verilog compatible or SDF-compatible simulator.
- VHDL – Generates a generic VHDL format netlist (*design*.vho) file and a VITAL VHDL (*design*.vto) file for use with any VHDL compatible simulator.
- VIEWLOGIC – Generates both a Viewlogic wir file (timing.1) to use with the Viewlogic’s ViewSim and PROsim timing simulators and a *design*.sim file for board-level simulation. In a Viewlogic design environment, the *design*.sim file is placed in your current working directory, and the timing.1 file is placed in your wir directory. This option can be used with TIMING_FILE to replace the “timing.1” default with a different file name. See the *pDS+ Fitter User Manual* for more information.

Device Programming

Once the design routes, the fusemap generation process of the pDS+ Fitter automatically reads the routed design information, converts the physical layout and routing of the design into device programming information, and generates a fusemap in standard JEDEC format. You can download the JEDEC (*design_name*.jed) file into an LSC ispLSI or pLSI device.

The JEDEC device programming file is used by any device programmer that accepts JEDEC fusemaps. For a list of LSC-compatible programmers, and further information on device programming options, contact your local LSC sales representative or see the *ISP Daisy Chain Download Reference Manual*.

Chapter 2 *pDS+ Fitter and Synario/ABEL Environment Setup*

This chapter describes how to install the pDS+ Synario/ABEL software. It also contains information on how to obtain software updates and contact the Lattice Semiconductor Corporation (LSC) customer support hotline.

System Requirements for pDS+ Synario/ABEL

The following sections list the software and hardware requirements for running pDS+ Synario/ABEL.

PC Software Requirements

The following software is required to run the pDS+ Synario/ABEL software on a PC:

- Data I/O Software – Synario, ABEL-WIN, or ABEL-DOS
- Lattice Semiconductor pDS+ Fitter (version 2.2 and above)
- SYN-pLSI Device Kit for Synario or ABEL-pLSI Device Kit for ABEL-WIN

PC Hardware Requirements

The following system configuration is required to run pDS+ Synario/ABEL on a PC:

- 386/486-based IBM compatible PC
- 16 megabytes of RAM
- Minimum 20 megabytes of free hard disk space
- 3-1/2” floppy disk drive
- One parallel printer port (for the software keys)
- Synario or ABEL security device

UNIX Workstation Software Requirements

The following software is required to run pDS+ ABEL on a UNIX workstation:

- Data I/O ABEL-SUN software
- The pDS+ Fitter (version 2.1 or above)

UNIX Workstation Hardware Requirements

The following system configuration is required to run pDS+ ABEL on a UNIX workstation:

- 3-button mouse
- SunOS version 4.x
- OpenWindows 3.0

Installing pDS+ Synario

This section describes the environment modifications for running the pDS+ Synario solution. See page 24 for the pDS+ Synario software and hardware requirements.

Synario Device Kit

You must install the Synario SYN-pLSI Device Kit, which is included in the LSC pDS+ Synario package, to access LSC ispLSI and pLSI devices. See the Data I/O *Getting Started User Manual* for information on installing and licensing your SYN-pLSI Device Kit.

Environment Setup

1. Install the Data I/O Synario software.
2. Install the LSC pDS+ Fitter. See the *pDS+ Fitter Installation Guide – PC* for installation instructions.
3. After installing the pDS+ Fitter, you must copy two files: `PLSI.DEV` and `ISPLSI.DEV` into <*dataio_install_path*>`\LIB5` or <*dataio_install_path*>`\LIB4` subdirectory manually. These two files can be found under the <*pdsplus_install_path*>`\ABEL` subdirectory.
4. Install the Synario SYN-pLSI Device Kit.

Additional Requirements

If you wish to perform Verilog functional or timing simulation, you must purchase and install a Verilog simulator. See the Data I/O *Verilog Simulator User Manual Synario Universal FPGA Design System* for further information.

Installing pDS+ ABEL-WIN

This section describes the environment modifications for running the pDS+ ABEL with the Data I/O ABEL-WIN software. See **page 24** for the pDS+ ABEL-WIN software and hardware requirements.

ABEL Device Kit

You must purchase the ABEL-pLSI Device Kit from Data I/O to access LSC ispLSI and pLSI devices. See the Data I/O *Getting Started User Manual* for information on installing and licensing your ABEL-pLSI Device Kit.

Environment Setup

1. Install the Data I/O ABEL-WIN software.
2. Install the LSC pDS+ Fitter. See the *pDS+ Fitter Installation Guide – PC* for installation instructions.
3. After installing the pDS+ Fitter, you must copy two files: `PLSI.DEV` and `ISPLSI.DEV` into `<`*`dataio_install_path`*`>\LIB5` or `<`*`dataio_install_path`*`>\LIB4` subdirectory manually. These two files can be found under the `<`*`pdsplus_install_path`*`>\ABEL` subdirectory.
4. Install the Data I/O ABEL-pLSI Device Kit.

Installing pDS+ ABEL-DOS

This section describes the environment modifications for running the pDS+ ABEL with the Data I/O ABEL-DOS software. See **page 24** for the pDS+ ABEL-DOS software and hardware requirements.

ABEL Memory Requirement

Data I/O provides three installation options with ABEL-DOS:

- ABEL standard memory version
- ABEL extended memory version
- ABEL mixed standard/extended memory version

Install the extended memory version of ABEL-DOS. With the standard memory version or the mixed standard/extended memory version, compiling large designs, or even designs with large test vector sets, can result in out-of-memory errors.

Environment Setup

1. Install the Data I/O ABEL-DOS software (extended memory version).
2. Install the LSC pDS+ Fitter. See the *pDS+ Fitter Installation Guide – PC* for installation instructions.
3. After installing the pDS+ Fitter, you must copy two files: `PLSI.DEV` and `ISPLSI.DEV` into <*`dataio_install_path`*>`\LIB5` or <*`dataio_install_path`*>`\LIB4` subdirectory manually. These two files can be found under the <*`pdsplus_install_path`*>`\ABEL` subdirectory.

Installing pDS+ ABEL-SUN

This section describes the environment modifications for running the pDS+ ABEL with the Data I/O ABEL-SUN software. See **page 24** for the pDS+ ABEL-SUN software and hardware requirements.

ABEL Memory Requirement

Data I/O provides three installation options with ABEL-SUN:

- ABEL standard memory version
- ABEL extended memory version
- ABEL mixed standard/extended memory version

Install the extended memory version of ABEL-SUN. With the standard memory version or the mixed standard/extended memory version, compiling large designs, or even designs with large test vector sets, can result in out-of-memory errors.

Environment Setup

1. Install the Data I/O ABEL-SUN software (extended memory version).
2. Install the pDS+ Fitter. See the *pDS+ Fitter Installation Guide – UNIX* for installation instructions.
3. Copy the `isplsi.dev` and `plsi.dev` files to the ABEL lib4 or lib5 directory as shown below.

```
cp /<pdsplus_install_path>/abel/*.dev
<path>/dataio4.3/abel/lib4
```

or

```
cp /<pdsplus_install_path>/abel/*.dev
<path>/dataio5.1/abel/lib5
```

Flex License Environment Configuration

After creating or editing the license.dat file, set the environment and path variables for your system. The following variable lines can typically be added to your .login or .cshrc file:

```
setenv LM_LICENSE_FILE <license_path>/license.dat
setenv PDSPLUS <install_path>
set path = ($path $PDSPLUS/bin)
```

Software Update Service

You will receive free software updates for one year if you fill out the License File Registration Form that is included with your software and fax it back to Lattice Semiconductor. Extended maintenance is available for purchase. Please contact your local Lattice Semiconductor sales representative for additional information.

Customer Hotline

If you have any questions or problems with this software, please call the Lattice Semiconductor Applications Hotline at **1-800-LATTICE** (1-800-528-8423) or (408) 428-6414. The Hotline is available from 8:00 AM to 5:00 PM, Pacific Time. Or, send e-mail to applications@latticesemi.com.

For Information On	Customer Resource	USA & Canada	Other Locations
ispLSI/pLSI Applications Support	Telephone Hotline	1-800-LATTICE	(408) 428-6414
	Fax	(408) 944-8450	
	Bulletin Board System	(408) 428-6417	
	E-mail	applications@latticesemi.com	
	FTP Site	http://www.latticesemi.com/ftp/index.html	
	World Wide Web	http://www.latticesemi.com	
GAL/ispGAL/ispGDS Applications Support	Telephone Hotline	1-800-FASTGAL	(503) 681-0118
	Fax	(503) 681-3037	
	Bulletin Board System	(503) 693-0215	
	E-mail	gal@latticesemi.com	
	FTP Site	http://www.latticesemi.com/ftp/index.html	
	World Wide Web	http://www.latticesemi.com	
Literature	Telephone Hotline	1-800-327-8425	(503) 681-0118
	Fax	(503) 681-3037	
	E-mail	gal@latticesemi.com	
	FTP Site	http://www.latticesemi.com/ftp/index.html	

Chapter 3 *Design Entry*

This chapter describes the design entry procedures for adding Lattice Semiconductor Corporation (LSC) Design Attributes to Synario schematic designs and ABEL-HDL designs. The following subjects are discussed:

- Design Attribute Overview
  - Entering Design Attributes in Synario Schematic Designs
  - Entering Design Attributes in ABEL-HDL Designs
  - Synario/ABEL-WIN Design Attribute Limitations in Hierarchical Designs
- ABEL-HDL Compatibility

Design Attribute Overview

Design Attributes control how your design is implemented into the logic resources of the target device. You can use Design Attributes in Synario schematics or in ABEL-HDL designs. Each Design Attribute you add, however, places restrictions on the pDS+ Fitter by giving it less freedom to use available logic resources. Apply these Design Attributes carefully to avoid overconstraining the compiler and possibly causing a routing failure. Table 3-1 provides a brief description of the Design Attributes. For further information, see the *pDS+ Fitter User Manual*.

Table 3-1. Design Attributes

Attribute	Attribute Placement	Attribute Description
CLK	Net	Assigns clock signals to specific clock lines.
GROUP	Net	Identifies GLB output grouping.
PRESERVE	Net	Prevents removal of a logic net during minimization and restricts optimization of your design.
SAP/EAP	Net	Specifies the Start and End of an Asynchronous Path. Asynchronous Path attributes prevent the router from duplicating GLB outputs.
SCP/ECP	Net	Specifies the Start and End of a Critical Path. Critical Path attributes restrict routing and can decrease resource utilization.
SNP/ENP	Net	Specifies the Start and End of a No-Minimize Path. No-Minimize Path attributes restrict design optimization.
LXOR2	Symbol (Node)	Enforces direct implementation of a two-input XOR.
PROTECT	Symbol (Node or Pin)	Prevents removal of a primitive during minimization.
REGTYPE	Symbol (Node or Pin)	Determines whether a register is placed in a GLB or an IOC.
CRIT	External Pin	Assigns specific outputs to use the ORP Bypass.
LOCK	External Pin	Assigns device I/O pins to signal names.
PULLUP	External Pin	Assigns specific IOCs to use pullup.
SLOWSLEW	External Pin	Assigns specific outputs to have slow slew rate.

Design Attributes in Synario Schematic Designs

The Data I/O Synario software allows you to enter designs as schematics, as ABEL-HDL descriptions, or as a combination of both. The following section describes how to add Design Attributes to Synario schematic designs. See page 34 for information on adding Design Attributes to Synario ABEL-HDL designs.

The following Synario schematic attributes have a corresponding LSC Design Attribute. All Design Attributes must be assigned to the top-level of your design or they will not be processed properly by the pDS+ Fitter.

Table 3-2 gives the LSC Design Attribute name and the corresponding Synario schematic attribute name. See the *pDS+ Fitter User Manual* for more information on Design Attributes. See the Data I/O *pLSI Device Kit Manual* for more information on Synario schematic attributes.

Table 3-2. LSC and Synario Schematic Design Attribute Names

LSC Attribute Name	**Synario Schematic Attribute Name**
CLK	Clock_type
CRIT	Critical
GROUP	Group
LOCK	SynarioPin
LXOR2	Use_XOR
PROTECT	Protect
PRESERVE	Preserve
PULLUP	PullUp
REGTYPE	Register_type
SCP/ECP	CriticalPath
SNP/ENP	NoMinimizePath
SAP/EAP	AsynchPath
SLOWSLEW	SlowSlew

Design Attributes in ABEL-HDL

The Data I/O Synario, ABEL-WIN, ABEL-DOS, and ABEL-SUN software allow you to enter Design Attributes in ABEL-HDL designs. Design Attributes are specified by including them in the top-level of the *design_name*.abl file in the form of pLSI property statements. The pLSI property statements allow you to pass control parameters to the pDS+ Fitter which are transparent to the Data I/O Synario/ABEL compiler.

The Design Attributes below are grouped functionally with an ABEL-HDL syntax example.

Net Attributes

- CLK

  Syntax:
  ```
  PLSI PROPERTY 'CLK pin_name CLK0|CLK1|CLK2|IOCLK0
              |IOCLK1|FASTCLK|SLOWCLK';
  ```

- GROUP

  Syntax:
  ```
  PLSI PROPERTY 'GROUP pin_name|node_name
               group_name';
  ```

- PRESERVE

  Syntax:
  ```
  PLSI PROPERTY 'PRESERVE node_name';
  ```

Path Attributes

- SAP/EAP

  Syntax:
  ```
  PLSI PROPERTY 'SAP pin_name|node_name path_name';
  PLSI PROPERTY 'SNP pin_name|node_name path_name';
  ```

- SCP/ECP

  Syntax:
  ```
  PLSI PROPERTY 'SCP pin_name|node_name path_name';
  PLSI PROPERTY 'ENP pin_name|node_name path_name';
  ```

- SNP/ENP

  Syntax:
  ```
  PLSI PROPERTY 'SNP pin_name|node_name path_name';
  PLSI PROPERTY 'ENP pin_name|node_name path_name';
  ```

Symbol Attributes

- LXOR2

  Syntax:
  `PLSI PROPERTY ’LXOR2` *`node_name`*`’;`

- PROTECT

  Syntax:
  `PLSI PROPERTY ’PROTECT` *`node_name`*`’;`

- REGTYPE

  Syntax:
  `PLSI PROPERTY ’REGTYPE` *`pin_name`*`|`*`node_name`* `GLB|IOC’;`

Pin Attributes

- CRIT

  Syntax:
  `PLSI PROPERTY ’CRIT` *`pin_name`*`’;`

- LOCK

  Syntax:
  `PLSI PROPERTY ’LOCK` *`pin_name pin_number`*`’;`

- PULLUP

  Syntax:
  `PLSI PROPERTY ’PULLUP` *`pin_name`*`’;`

- SLOWSLEW

  Syntax:
  `PLSI PROPERTY ’SLOWSLEW` *`pin_name`*`’;`

Synario/ABEL-WIN Design Attribute Restrictions in Hierarchical Designs

The current versions of Data I/O Synario and ABEL-WIN do not properly process Design Attributes in the lower-level modules of hierarchical designs. Only place Design Attributes and Compiler Control Options in the top-level of your design. All Design Attributes and Compiler Control Options must be placed in the top-level of your design because Synario and ABEL-WIN do not support parameterized attributes.

A parameterized attribute is any attribute that contains a variable. Use parameterized attributes when there is more than one instance of a macro in a design and you wish to assign a different attribute value to each instance. Parameterized attributes can be placed in different instances on the same level of your design, or in different hierarchical levels. Although the pDS+ Fitter accepts parameterized attributes, the Data I/O Synario/ABEL-WIN software does not support the variable syntax required to implement them. The ABEL-HDL hierarchical design example is included for reference purposes only.

ABEL-HDL Hierarchical Design Example

In the following design example, the first module titled “top,” represents the top-level of a hierarchical design. The second module, titled “bottom,” represents the lower-level of the hierarchical design. After the design is processed and compiled by the ABEL compiler, the design is flattened. When the design is flattened, the net names, “a1” and “b1,” in the lower-level module titled “bottom,” become unique for each instantiation of the module. The path names do not become unique after the design is flattened; this produces an error.

Although the following example is shown in ABEL-HDL, the same problem exists in Synario schematic designs. Design Attributes and Compiler Control Options, therefore, must be applied to the top-level of your design for them to be processed correctly.

```
MODULE top

"inputs
    in11, in12, in13       pin;
    in14, in15             pin;

"outputs

    out11, out12           pin;
    out21, out22           pin;


"sub-module declarations

bottom interface (a1, a2, a3, a4, a5 -> b1, b2);

"sub-module instances

nomin1 functional_block bottom;
nomin2 functional_block bottom;

equations

    nomin1.a1 = in11;
    nomin1.a2 = in12;
    nomin1.a3 = in13;
    nomin1.a4 = in14;
    nomin1.a5 = in15;
    out11 = nomin1.b1;
    out12 = nomin1.b2;

    nomin2.a1 = in11;
    nomin2.a2 = in12;
    nomin2.a3 = in13;
    nomin2.a4 = in14;
    nomin2.a5 = in15;
    out21 = nomin2.b1;
    out22 = nomin2.b2;
```

```
MODULE bottom

interface (a1,a2, a3, a4, a5 -> b1, b2);

"inputs

    a1, a2, a3, a4, a5  pin;

    d1, e1              node;
    d2, e2              node;

"output
    b1, b2              pin;

    plsi property 'snp a1 path1';
    plsi property 'enp b1 path1';

equations
    d1 = a1 # a2;
    e1 = a2 # a3;
    b1 = d1 $ e1;

    d2 = a3 # a4;
    e2 = a4 # a5;
    b2 = d2 $ e2;
```

ABEL-HDL Compatibility

The following sections describe the compatibility of the pDS+ Fitter with ABEL-HDL.

ABEL–HDL Dot Extensions and Mapping

Table 3-3 lists the ABEL dot extensions supported by LSC architecture and the functions to which they map. Figure 3-1 shows how these dot extensions map to device inputs and outputs.

Table 3-3. ABEL Dot Extensions and Mapping

Dot Extensions	**Maps to . . .**
.RE, .AR	Asynchronous reset input of D-type flip-flop. This is not the Global Reset which is always connected.
.AP, .PR	Asynchronous register preset.
.J, .K, .T, .S, .R	Logic in front of D-type flip-flop creating a JK, T, or SR flip-flop.
.SP, .SR, .SET, .CLR	Synchronous preset and synchronous reset logic input to D-type flip-flops.
.ACLR, .ASET	Synchronous/asynchronous register reset and preset.
.D	Data input of D-type flip-flop or latch.
.Q	Register feedback.
.LE	Enable input of a transparent low latch.
.LH	Enable input of a transparent high latch.
.CLK	Clock input of any flip-flop.
.OE	Output enable of 3-state driver.
.FB	Register feedback, normalized to pin polarity.
.PIN	Feedback from I/O pin.
.CE	Clock enable logic of a D-type flip-flop.

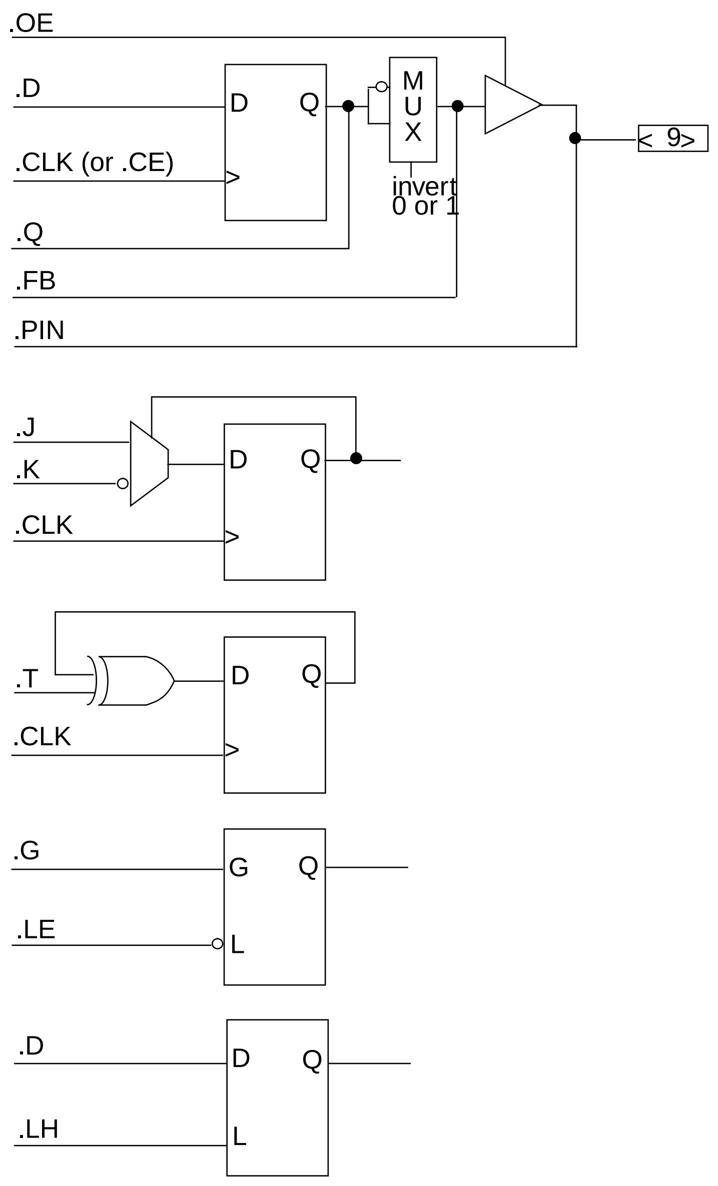

Figure 3-1. Dot Extension Mapping

NOTE

Use .FB for register feedback, normalized to pin polarity;
.FB is only valid for register outputs declared as pins.
For registers declared as nodes, use .Q for register feedback and use ISTYPE “REG_D,” ISTYPE “REG_T,” and so on, instead of the ISTYPE “REG” statement.

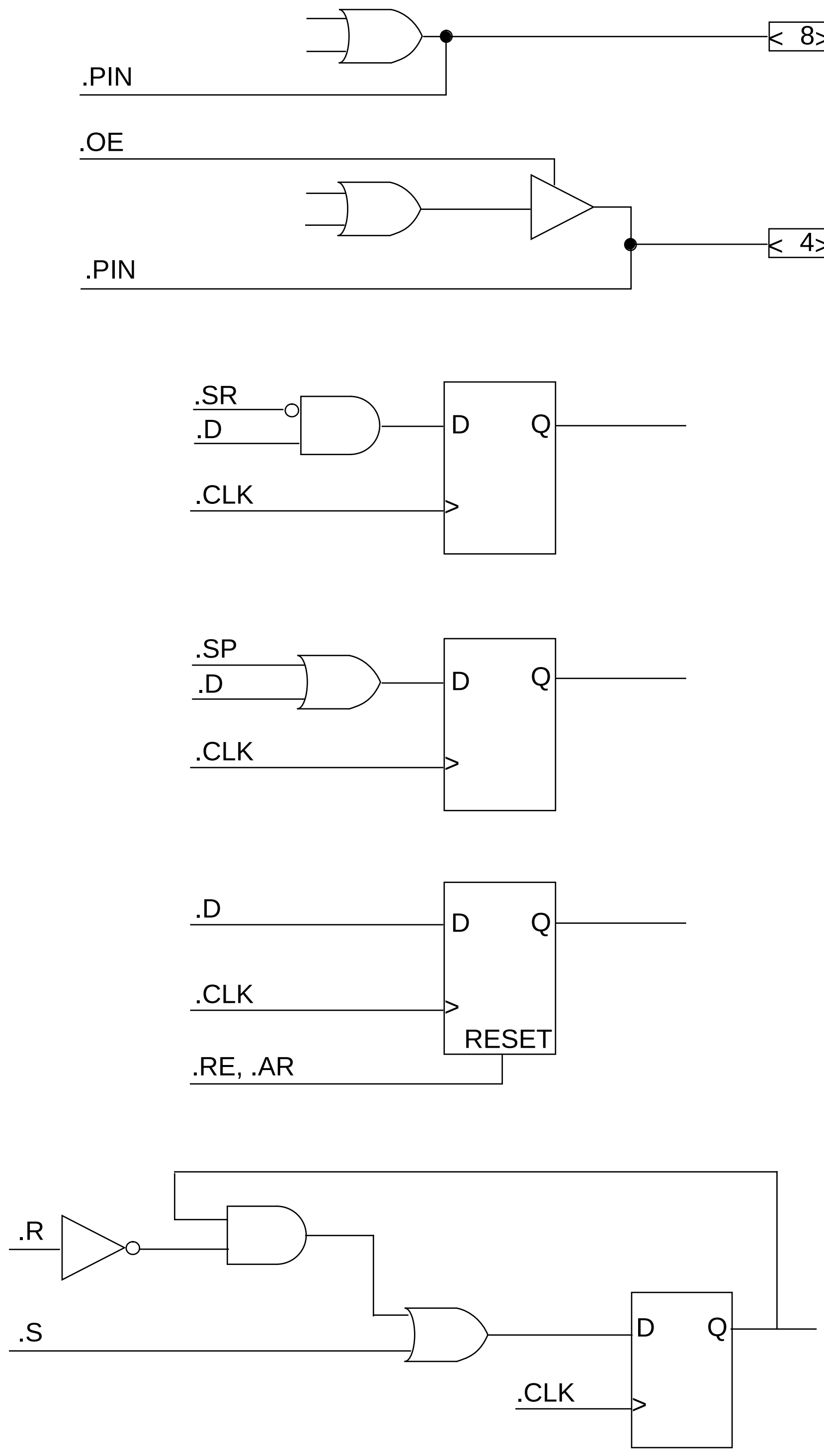

Figure 3-1. Dot Extension Mapping (Continued)

ABEL Signal Attributes

Attributes are assigned to signals using the ABEL ISTYPE statement in the Declaration section of the ABEL source file. Table 3-4 shows these attributes. For more information on ABEL signal attributes, see the *ABEL Design Software User Manual*.

Table 3-4. ABEL Signal Attributes

Attribute	Description
buffer	Non-inverting output pin.
com	Combinational signal.
invert	Inverted registered output pin.
neg	Complement signal before processing in ABEL compilation.
pos	Do not complement signal before processing in ABEL compilation.
reg	Generic flip-flop.
reg_D	D-type flip-flop.
reg_T	T-type flip-flop.
reg_G	D-type flip-flop with a gated clock.
reg_JK	JK-type flip-flop.
reg_SR	SR-type flip-flop.
XOR	Retains XOR notation instead of AND or OR implementation. XOR will use LXOR2.

ABEL-HDL Exceptions

The following ABEL-HDL statements are ignored because they do not apply to pDS+ Synario/ABEL:

- FUSES – The LSC FUSEGEN program is used instead.
- XOR_Factors – The LSC PLSI PROPERTY LXOR2 is used instead.

Chapter 4 *Design Compilation*

This chapter describes how to add Lattice Semiconductor Compiler Control Options, how to specify an LSC device, and how to invoke the pDS+ Fitter. This chapter contains information on the following topics.

- Compiler Control Options Overview
- Synario and the pDS+ Fitter
- ABEL-WIN and the pDS+ Fitter
- ABEL-DOS and the pDS+ Fitter
- ABEL-SUN and the pDS+ Fitter

Compiler Control Options Overview

The following Compiler Control Options determine how your design is partitioned, placed, and routed into the physical resources of the target device. Apply these options carefully to avoid overconstraining the compiler and possibly causing a compiler failure. Table 4-1 provides a brief description of each Compiler Control Option. See the *pDS+ Fitter User Manual* for further details.

Table 4-1. Compiler Control Options

Attribute	Attribute Description
CARRY_PIN_DIRECTION	Attempts to maintain user-specified pin directions for 3-state outputs into any simulation output netlist.
CASE_SENSITIVE	Enables the compiler to treat identifiers, such as pin names and net names, as case-sensitive or case-insensitive.
EFFORT	Provides a number of different optimization options ranging from 1 to 3.
EXTENDED_ROUTE	Instructs the router to use a complete routing cycle in an attempt to route a design.
IGNORE_FIXED_PIN	Instructs the compiler to either ignore (ON) or honor (OFF) the pin locking attributes in your design.
MAX_GLB_IN	Specifies the maximum number of GLB inputs (ranging from 2 to 24, depending on the device family) the compiler is allowed to use for each GLB in your design.
MAX_GLB_OUT	Specifies the maximum number of GLB outputs (ranging from 1 to 4) the compiler is allowed to use for each GLB in your design.
OUTPUT_FORM	Specifies the output netlist format to be generated for post-route simulation. Values are EDIF, LDF, ORCAD, PREROUTE_LDF, SIM, VERILOG, VHDL, and VIEWLOGIC. There is no default for this option.
PARAM_FILE	Allows Compiler Control Options to be read from a file other than the ABEL source file.
PART	Specifies the part number of the target device.
PIN_FILE	Directs the compiler to read the pin file for pin assignments.

Table 4-1. Compiler Control Options (Continued)

STRATEGY	Allows you to specify one of the following optimization strategies: AREA, DELAY, or NO_OPTIMIZATION.
TIMING_FILE	Directs the compiler to generate a Viewlogic timing simulation wir file (*file_name*.1) with the OUTPUT_FORM VIEWLOGIC option.
USE_GLOBAL_RESET	Frees the global reset pin for use by the compiler.
ISP	Informs the software that you want to use the ISP pins on an ispLSI device.
ISP_EXCEPT_Y2	Allows the software to reserve the Y2 clock input for routing in an ispLSI device.
PULLUP	Specifies the use of I/O pin pull-up resistors to the compiler.
SECURITY	Influences the device security cell programming.
Y1_AS_RESET	Determines the Y1 pin use on ispL/pLSI 1016 and 2032 devices.

ABEL-HDL Syntax Examples

The syntax examples below present ABEL-HDL source file syntax. See the *pDS+ Fitter User Manual* for a syntax example of how to enter Compiler Control Options in the Parameter File or from the command line.

- CARRY_PIN_DIRECTION

  Syntax:
  ```
  PLSI PROPERTY ’CARRY_PIN_DIRECTION ON|OFF’;
  ```

- CASE_SENSITIVE

  Syntax:
  ```
  PLSI PROPERTY ’CASE_SENSITIVE ON|OFF’;
  ```

- EFFORT

  Syntax:
  ```
  PLSI PROPERTY ’EFFORT 1|2|3’;
  ```

- EXTENDED_ROUTE

  Syntax:
  ```
  PLSI PROPERTY ’EXTENDED_ROUTE ON|OFF’;
  ```

- IGNORE_FIXED_PIN

  Syntax:
  ```
  PLSI PROPERTY ’IGNORE_FIXED_PIN ON|OFF’;
  ```

- MAX_GLB_IN

  Syntax:
  ```
  PLSI PROPERTY ’MAX_GLB_IN 2|..|24’;
  ```

- MAX_GLB_OUT

  Syntax:
  ```
  PLSI PROPERTY ’MAX_GLB_OUT 1|2|3|4’;
  ```

- OUTPUT_FORM

  Syntax:
  ```
  PLSI PROPERTY ’OUTPUT_FORM EDIF|LDF|ORCAD|SIM|
    PREROUTE_LDF|VERILOG|VHDL|VIEWLOGIC’;
  ```

- PARAM_FILE

  Syntax:

  `PLSI PROPERTY ’PARAM_FILE `*`file_name`*`’;`

- PART

  Syntax:

  `PLSI PROPERTY ’PART `*`_part_name`*`’;`

  See the *pDS+ Fitter Release Notes* for a complete list of the available LSC part names.

- PIN_FILE

  Syntax:

  `PLSI PROPERTY ’PIN_FILE `*`file_name`*`’;`

- STRATEGY

  Syntax:

  `PLSI PROPERTY ’STRATEGY AREA|DELAY’;`

- TIMING_FILE

  Syntax:

  `PLSI PROPERTY ’TIMING_FILE `*`file_name`*`’;`

- USE_GLOBAL_RESET

  Syntax:

  `PLSI PROPERTY ’USE_GLOBAL_RESET ON|OFF’;`

- Device Control Options
  - ISP

Syntax:

```
PLSI PROPERTY ’ISP ON|OFF’;
```

  - ISP_EXCEPT_Y2

Syntax:

```
PLSI PROPERTY ’ISP_EXCEPT_Y2 ON|OFF’;
```

  - PULLUP

Syntax:

```
PLSI PROPERTY ’PULLUP ON|OFF’;
```

  - SECURITY

Syntax:

```
PLSI PROPERTY ’SECURITY ON|OFF’;
```

  - Y1_AS_RESET

Syntax:

```
PLSI PROPERTY ’Y1_AS_RESET ON|OFF’;
```

Synario and the pDS+ Fitter

Specifying Compiler Control Options

The Compiler Control Options you specify through the Synario Fit Process Properties Editor menu or the Parameter File (*file_name*.par) determine how your design is partitioned, placed, and routed into the physical resources of the target device. Use the Synario Fit Process Properties Editor to select Compiler Control Options and to pass the name of the Parameter File to the pDS+ Fitter. If the Parameter File name specified in the Synario Fit Process Properties Editor is not present, the pDS+ Fitter issues a warning and continues processing using default settings. Table 4-2 lists the Compiler Control Options and the corresponding Synario menu Compiler Control Option.

Table 4-2. LSC and Synario Compiler Control Options

LSC Compiler Control Options	**Synario Fitter Control Options**
CASE_SENSITIVE	Use Case Sensitivity
EFFORT	Fit Effort
EXTENDED_ROUTE	Extended Route
IGNORE_FIXED_PIN	Ignore Fixed Pins
MAX_GLB_IN	Maximum GLB Inputs
MAX_GLB_OUT	Maximum GLB Outputs
OUTPUT_FORM SIM	Generate SIM File
OUTPUT_FORM VERILOG	Generate Verilog Timing Model
PARAM_FILE	Parameter File Name
PIN_FILE	Pin File Name
STRATEGY	Fit Strategy
USE_GLOBAL_RESET	Use Global Reset

Invoking the pDS+ Fitter

Within the Synario design environment, the pDS+ Fitter runs automatically when you select a valid LSC device name.

To specify an LSC device in Synario:

1. Double-click **Virtual Device** under the **Sources** heading in the Synario Project Navigator window.
2. Click on the **pLSI** option in the Choose Device dialog box.
3. Double-click **Fit Design** to invoke the pDS+ Fitter.

Controlling the pDS+ Fitter

When using the Synario design environment, the following is the order of precedence:

- Synario Fit Process Properties Editor menu
- Parameter File
- pLSI Property Statements in the ABEL-HDL source file

Specifying the Parameter File

Specify the Parameter File in Synario through the Fit Process Properties Editor menu. When the pDS+ Fitter is invoked, it searches for this file and reads its contents. It performs fitting based on the options listed. If this file is not present, the pDS+ Fitter issues a warning and continues processing based on default options.

Synario Design Example

The following design example demonstrates the steps used to take a schematic design created in Synario from design entry through device programming. The schematic shown in Figure 4-1 is the source for the example.

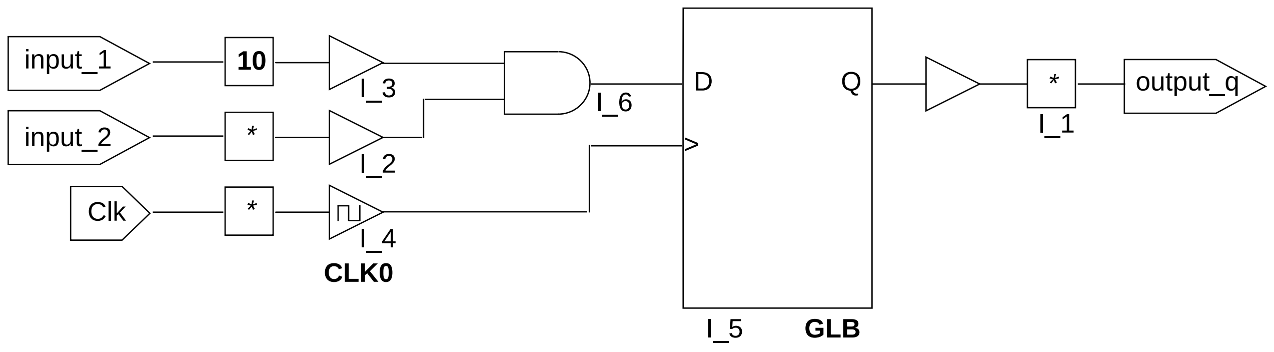

Figure 4-1. andff.sch Schematic Design Example

To download a device using a Synario Schematic Design:

1. Create an andff project directory.
2. Create an andff.sch (schematic) design.
3. Create an andff.tf (test fixtures) file.
4. Perform functional simulation (optional).
5. Double-click **Virtual Device** under the **Sources** heading in the Synario Project Navigator window.
6. Double-click the **pLSI** option in the Choose Device dialog box.
7. Double-click on a device name in the **Device Field** of the Choose Device dialog box to select a device. For the purpose of this example, select **ispLSI 1016-60 JLCC44/883** and click **OK**.
8. Click **Yes** in the Confirm Change dialog box.
9. Add the desired LSC Design Attributes.
   a. Double-click on the **andff.sch** file under the **Sources in Project** section of the Synario Project Navigator window. The Schematic Editor window appears.
   b. Select **Add** ⇒ **Symbol Attribute** and the Symbol Attribute Editor window appears. Select the I_5 symbol (which is a flip-flop) and the available attributes appear in the window. Click on **Register Type** and type `GLB` in the field box, then click **Close**. (The attribute will not appear on the schematic. It is added to the schematic design in Figure 4-1 for reference purposes only.)

c. Select **Add** ⇒ **Symbol Attribute** and the Symbol Attribute Editor window appears. Select the I_4 symbol (which is a clock) and the available attributes appear in the window. Click on **Clock_type** and type **`CLK0`** in the field box, then click **Close**. (The attribute will not appear on the schematic. It is added to the schematic design in Figure 4-1 for reference purposes only.)

d. Select **Add** ⇒ **Symbol Attribute** and the Symbol Attribute Editor window appears. Select the I_3 symbol (which is a pin) and the available attributes appear in the window. Click on **SynarioPin** and type **`10`** in the field box, then click **Close**. (The pin attribute, in this case, will appear as shown on the schematic in Figure 4-1.)

e. Select **File** ⇒ **Save**, then click **Close**.

10. Select **Synario Fit Process Properties Editor**. Click on the **ispLSI 1016-60 JLCC 44/883** LSC part name in the Synario Project Navigator menu. Single-click on **Fit Design**. Click on the **Properties** button at the bottom of the window.
11. Generate a Verilog model for timing simulation by either entering the name of your Parameter File, *file_name*.par (which contains the **`OUTPUT_FORM verilog`** statement), or by selecting **Generate Verilog Timing Model** and specifying the **Yes** option.
12. Select any other Compiler Control Options you wish to specify in your design. Click **Close**.
13. Double-click **Fit Design** to invoke the pDS+ Fitter.
14. Perform timing simulation (optional).
15. Program an LSC device.

ABEL-WIN and the pDS+ Fitter

The Compiler Control Options you specify through ABEL-WIN with the ABEL Fit Process Properties Editor menu or the Parameter File (*file_name*.par) determine how your design is partitioned, placed, and routed into the physical resources of the target device.

Use the ABEL Fit Process Properties Editor to select Compiler Control Options and to pass the name of the Parameter File to the pDS+ Fitter. If the Parameter File name specified in the ABEL Fit Process Properties Editor is not present, the pDS+ Fitter issues a warning and continues processing using default settings. Table 4-3 lists the LSC Compiler Control Options and the corresponding ABEL-WIN menu Compiler Control Option.

Table 4-3. LSC and ABEL Compiler Control Options

LSC Compiler Control Options	**ABEL-WIN Fitter Control Options**
CASE_SENSITIVE	Use Case Sensitivity
EFFORT	Fit Effort
EXTENDED_ROUTE	Extended Route
IGNORE_FIXED_PIN	Ignore Fixed Pins
MAX_GLB_IN	Maximum GLB Inputs
MAX_GLB_OUT	Maximum GLB Outputs
OUTPUT_FORM SIM	Generate SIM File
PARAM_FILE	Parameter File Name
PIN_FILE	Pin File Name
STRATEGY	Fit Strategy
USE_GLOBAL_RESET	Use Global Reset

Invoking the pDS+ Fitter

Within the ABEL-WIN design environment, the pDS+ Fitter runs automatically when you select a valid LSC device name.

To specify an LSC device in ABEL-WIN:

1. Double-click **Virtual Device** under the **Sources** heading in the ABEL Project Navigator window.
2. Click on the **pLSI** option in the Choose Device dialog box.
3. Double-click **Fit Design** to invoke the pDS+ Fitter.

Controlling the pDS+ Fitter

When using the ABEL-WIN design environment, the following is the order of precedence:

- ABEL Fit Process Properties Editor menu
- Parameter File
- PLSI Property Statements in the ABEL-HDL source file

Specifying the Parameter File

Specify the Parameter File in ABEL-WIN through the Fit Process Properties Editor menu. When the pDS+ Fitter is invoked, it searches for this file and reads its contents. It performs fitting based on the options listed. If this file is not present, the pDS+ Fitter issues a warning and continues processing based on default options.

ABEL-WIN Design Example

This design example demonstrates the steps used to take a design created in ABEL-WIN from design entry through device programming. The ABEL-HDL source file shown in Figure 4-2 is the source for the following ABEL-WIN design example.

```
MODULE andff

TITLE ‘And gate and flip-flop’

input_1, input_2, Clkpin;
output_q           pin istype ‘reg’;

plsi property ’regtype output_q glb’;
plsi property ’lock input_1 10’;
plsi property ’clk Clk clk0’;

Equations

output_q   := input_1 & input_2;
output_q.clk = Clk;

END
```

Figure 4-2. andff.abl ABEL-WIN Design Example

To download a device using an ABEL-WIN Design:

1. Create an andff project directory.
2. Create an andff.abl (ABEL-HDL) design.
3. Create an andff.tf (test fixture) file.
4. Perform functional simulation (optional).
5. Double-click **Virtual Device** under the **Sources** heading in the ABEL Project Navigator window. Double-click the **pLSI** option in the Choose Device dialog box.
6. Double-click on a device name in the **Device Field** of the Choose Device dialog box to select a device. For the purpose of this example, select **ispLSI 1016-60 JLCC44/883** and click **OK**.
7. Click **Yes** in the Confirm Change dialog box.

8. Add the desired LSC Design Attributes to your andff.abl file. For example:

   plsi property ’regtype output_q glb’;

   plsi property ’lock input_1 10’;

   plsi property ’clk Clk clk0’;

9. Select **ABEL Fit Process Properties Editor**. Click on the **ispLSI 1016-60 JLCC 44/883** LSC part name in the ABEL Project Navigator menu. Single-click on **Fit Design**. Click on the **Properties** button at the bottom of the window.
10. Select any other Compiler Control Options you wish to specify in your design. Click **Close**.
11. Double-click **Fit Design** to invoke the pDS+ Fitter.
12. Perform a timing simulation (optional).
13. Program an LSC device.

ABEL-DOS and the pDS+ Fitter

Some ABEL-DOS SmartPart Options are supported by pDS+ ABEL; these SmartPart Options are listed with their Parameter File equivalents in Table 4-4.

The ABEL SmartPart Options override the properties specified during design entry using LSC pLSI Property Statements, or any specifications in the Parameter File. The SmartPart Options are not stored, but are shown in your log file or report file.

See *ABEL-HDL Syntax Examples* on page 47 for information on how to enter Compiler Control Options in your ABEL-HDL design. See the *pDS+ Fitter User Manual* for examples of how to enter Compiler Control Options in your Parameter File or from the command line.

Supported SmartPart Options

The LSC Compiler Control Options which have a corresponding ABEL-DOS menu equivalent are listed below.

Table 4-4. LSC Compiler Control Options and the ABEL-DOS Equivalent

LSC Compiler Control Option	ABEL-DOS Menu	ABEL-DOS Menu Description
DEVICE	**SmartPart ⇒ Device**	Enter the LSC device family name, ispLSI or pLSI, in the Device field.
IGNORE_FIXED_PIN ON	**SmartPart ⇒ Ignore Preassignments**	This option instructs the compiler to ignore any pin assignments specified during design entry using the ABEL pin statement, or the LOCK property.
STRATEGY AREA	**SmartPart ⇒ Optimize Fit for Area**	This option attempts to create the smallest design (without considering speed).
STRATEGY DELAY	**SmartPart ⇒ Optimize Fit for Speed**	This option attempts to create the fastest design (without considering area).

Invoking the pDS+ Fitter

Within the ABEL-DOS environment, the pDS+ Fitter runs automatically when you specify the following:

- A valid LSC device name
- **SmartPart** ⇒ **FIT** under the ABEL-DOS menu

To specify an LSC device using ABEL-DOS:

1. Add a device declaration statement to your ABEL-DOS source file (*design_name*.abl). Specify either ispLSI or pLSI.

   Syntax:

   ```
   device_id DEVICE 'LSC_device_family';
   ```

   Example:

   ```
   u1 DEVICE 'pLSI';
   ```

Or specify a pLSI or an ispLSI device in the **SmartPart** ⇒ **Options** dialog box in the ABEL-DOS Design Environment Menu.

2. Enter a pLSI PROPERTY statement 'PART *_part_name'* in the ABEL-HDL source code.

   Syntax:

   ```
   PLSI PROPERTY 'PART part_name';
   ```

   Example:

   ```
   PLSI PROPERTY 'PART pLSI1048C-70LQ128';
   ```

Or enter a pLSI PROPERTY statement 'PART *part_name'* in the Parameter File.

✍ ***NOTE*** If this property statement is missing, the default part is: ispLSI1032E-90LT100.

Controlling the pDS+ Fitter

Within the ABEL-DOS design environment, the following is the order of precedence:

- The ABEL-DOS **SmartPart** ⇒ **Options** menu overrides the Parameter File and the pLSI Property Statements in an ABEL-HDL source file. The SmartPart menu develops statements that are equivalent to the pLSI Property Statements. Only a limited number of these pLSI Property Statements are available in the ABEL-DOS SmartPart Option Menu.
- The Parameter File overrides all pLSI Property Statements.
- The pLSI Property Statements in the ABEL-HDL source file override any defaults.

Specifying the Parameter File

A Parameter File is specified in ABEL-DOS by placing a “PLSI PROPERTY PARAM_FILE *file_name”* statement in your source file.

ABEL Design Example

This design example demonstrates the steps used to take a design created in ABEL-DOS from design entry through device programming. The example uses the attribute test circuit count5.abl from the\pdsplus_install_path\abel\examples directory.

1. Enter your design description in ABEL-HDL format. The following is an example:

```
"constants
      C,X,Z,P,H,L   = .C., .X., .Z., .P., 1, 0;
        COUNT = [Q2,Q1,Q0];

" Counter States
        S0 = ^b000;    S4 = ^b100;
        S1 = ^b001;    S5 = ^b101;
        S2 = ^b010;    S6 = ^b110;
        S3 = ^b011;    S7 = ^b111;

equations
        COUNT.CLK= Clock;
        COUNT.OE = !OutEn;

state_diagram COUNT
        State S0:       IF !Clear THEN S1 ELSE S0;

        State S1:       IF !Clear THEN S2 ELSE S0;

        State S2:       IF !Clear THEN S3 ELSE S0;

        State S3:       IF !Clear THEN S4 ELSE S0;

        State S4:       GOTO S0;
```

```
"Return from illegal state, reset counter to 0.
        State S5:  GOTO S0;
        State S6:  GOTO S0;
        State S7:  GOTO S0;

test_vectors 'Test Counter'
      ( [Clock, OutEn, Clear ] -> COUNT)
        [ C,      0,     1   ] ->   0;
       "Simulation to indicate output value.
        [ C,      0,     0   ] -> ^b001;
       "Define output as a binary value.
        [ C,      0,     0   ] ->   2;
       "Define output as a decimal value.
        [ C,      0,     0   ] ->  ^h3;
       "Define output as a hex value.
        [ C,      0,     0   ] ->  S4;
       "Define output as a set.
        [ C,      0,     0   ] ->  S0;
        [ C,      1,     0   ] ->   Z;
       "Define output as tristate.
        [ C,      0,     0   ] ->  S2;
        [ 0,      0,     0   ] ->  S2;
        [ C,      0,     0   ] ->  S3;
        [ C,      0,     0   ] ->  S4;
        [ C,      0,     0   ] ->  S0;
        [ C,      0,     0   ] ->  S1;
        [ C,      0,     0   ] ->  S2;
        [ C,      0,     1   ] ->  S0;

test_vectors 'Preload counter to invalid states'
" NOTE: The ABEL preload function assumes an inversion
"between the register outputs and device outputs. Register
"PRELOAD function can only be use to do functional
"simulation. The pLSI parts do not support register
"preloads.
      ( [Clock, OutEn, Clear, COUNT] -> COUNT)
        [ P,      1,     0,    S1 ] ->   X  ;
       "ACTUAL PRELOAD=S6
        [ 0,      0,     0,     X ] ->   S6 ;
        [ C,      0,     0,     X ] ->   S0 ;
        [ P,      1,     0,    S0 ] ->   X  ;
        "ACTUAL PRELOAD=S7
        [ 0,      0,     0,     X ] ->   S7 ;
        [ C,      0,     0,     X ] ->   S0 ;
end
```

2. Add LSC Design Attributes, Compiler Control Options, and specify an ispLSI or pLSI device in the ABEL-HDL source file. The following is an example:

```
".......................INPUTS.........................."
        Clock,Clear,OutEn      pin ;

"..........................OUTPUTS........................."
        Q2,Q0                pin 16,17 istype 'reg_D,buffer';
        Q1                   PIN ;
        Q1                   ISTYPE 'REG_D,BUFFER';

"..........................OPTIONAL........................"
        PLSI PROPERTY 'LOCK Q1 15 ';
        PLSI PROPERTY 'CLK Clock FASTCLK';
        PLSI PROPERTY 'Y1_AS_RESET ON';
        PLSI PROPERTY 'PART ispLSI1016-110LJ44';
        PLSI PROPERTY 'STRATEGY AREA';

"..........................................................."
```

3. Compile the ABEL-HDL source file to produce a .tt2 file using the ABEL compiler.
4. Compile the .tt2 file with the pDS+ Fitter.
5. Program the specified LSC device.

ABEL-SUN and the pDS+ Fitter

Some ABEL-SUN SmartPart Options are supported in pDS+ ABEL; these SmartPart Options are listed with their Parameter File equivalents in Table 4-4.

The ABEL-SUN SmartPart Options override the properties specified during design entry using LSC pLSI Property Statements, or any specifications in the Parameter File. The SmartPart Options are not stored, but are shown in your log or report file.

See *ABEL-HDL Syntax Examples* on page 47 for information on how to enter Compiler Control Options in your ABEL-HDL design. See the *pDS+ Fitter User Manual* for information on how to enter Compiler Control Options in your Parameter File or from the command line.

Supported SmartPart Options

The LSC Compiler Control Options which have a corresponding ABEL-SUN menu equivalent are listed below.

Table 4-5. LSC Compiler Control Options and the ABEL-SUN Equivalent

LSC Compiler Control Option	ABEL-SUN Menu	ABEL-SUN Menu Description
DEVICE	**SmartPart ➠ Device**	Enter the LSC device family name, ispLSI or pLSI, in the Device field.
LOCK	**SmartPart ➠ Attempt Preassignments**	This option instructs the compiler to try all the pin assignments specified during design entry using the ABEL pin statement, or the LOCK property.
IGNORE_FIXED_PIN ON	**SmartPart ➠ Ignore Preassignments**	This option instructs the compiler to ignore any pin assignments specified during design entry using the ABEL pin statement, or the LOCK property.
LOCK	**SmartPart ➠ Keep Preassign-ments**	This option instructs the compiler to keep the pin assignments specified during design entry using the ABEL pin statement, or the LOCK property. If the existing pin assignments do not fit into the selected device, the routing will fail.
STRATEGY AREA	**SmartPart ➠ Optimize Fit for Area**	This option attempts to create the smallest design (without considering speed).

Table 4-5. LSC Compiler Control Options and the ABEL-SUN Equivalent (Continued)

LSC Compiler Control Option	ABEL-SUN Menu	ABEL-SUN Menu Description
STRATEGY DELAY	**SmartPart ⇒ Optimize Fit for Speed**	This option attempts to create the fastest design (without considering area).

Invoking the pDS+ Fitter

ABEL-SUN 4.3

Within the ABEL-SUN 4.3 environment, the pDS+ Fitter runs automatically when you specify the following:

- A valid LSC device name
- **SmartPart** ⇒ **FIT** under the ABEL-SUN menu

To specify an LSC device using ABEL-SUN (4.3):

1. Add a device declaration statement to your ABEL-SUN source file (*design_name*.abl). Specify either ispLSI or pLSI.

   Syntax:

   ```
   device_id DEVICE 'LSC_device_family';
   ```

   Example:

   ```
   u1 DEVICE 'pLSI';
   ```

   Or specify a pLSI or an ispLSI device in the **SmartPart** ⇒ **Options** dialog box in the ABEL-SUN Design Environment Menu.

2. Enter a pLSI PROPERTY statement 'PART *part_name'* in the ABEL-HDL source code.

   Syntax:

   ```
   PLSI PROPERTY 'PART part_name';
   ```

   Example:

   ```
   PLSI PROPERTY 'PART pLSI1048C-70LQ128';
   ```

   Or enter a pLSI PROPERTY statement 'PART *part_name'* in the Parameter File.

✍ ***NOTE*** If this property statement is missing, the default part is: ispLSI1032E-90LT100.

ABEL-SUN 5.1

Within the ABEL-SUN 5.1 design environment, pDS+ ABEL runs automatically when you specify the following:

- A valid LSC device name
- **Fit** ⇒ **Fit** under the ABEL-SUN menu

To specify an LSC device using ABEL-SUN (5.1):

1. Add a device declaration statement to your ABEL source file (*design_name*.abl). Specify either ispLSI or pLSI.

   Syntax:

   ```
   device_id DEVICE 'LSC_device_family';
   ```

   Example:

   ```
   u1 DEVICE 'pLSI';
   ```

   Or specify an ispLSI or a pLSI device in the **Options** ⇒ **SmartPart** dialog box. See Chapter 3, "Design Entry."

   Or specify an ispLSI or a pLSI device in the **Options** ⇒ **Fit...** dialog box. See Chapter 3, "Design Entry."

2. Enter a pLSI Property Statement 'PART *part_name'* in the ABEL-HDL source code.

   Syntax:

   ```
   PLSI PROPERTY 'PART part_name';
   ```

   Example:

   ```
   PLSI PROPERTY 'PART pLSI1048C-70LQ128';
   ```

Or enter a pLSI Property Statement 'PART *part_name'* in the Parameter File (*file_name*.par).

NOTE If this property statement is missing, the default part is: ispLSI1032E-90LT100.

Controlling the pDS+ Fitter

ABEL-SUN 4.3

Within the ABEL-SUN 4.3 design environment, the following is the order of precedence:

- The ABEL-SUN **SmartPart** ⇒ **Options** menu overrides the Parameter File and the pLSI Property Statements in an ABEL-HDL source file. The SmartPart menu develops statements that are equivalent to the pLSI Property Statements. Only a limited number of these pLSI Property Statements are available in the ABEL-SUN SmartPart Option menu.
- The Parameter File overrides all pLSI Property Statements.
- The pLSI Property Statements in the ABEL-HDL source file override any defaults.

ABEL-SUN 5.1

Within the ABEL-SUN 5.1 design environment, the following is the order of precedence:

- The ABEL-SUN **Options** ⇒ **SmartPart...** menu overrides the pLSI Property Statements in an ABEL-HDL source file and in the Parameter File.
- The ABEL-SUN **Options** ⇒ **Fit...** menu overrides the Parameter File and the pLSI Property Statements in an ABEL-HDL source file.
- The Parameter File overrides all pLSI Property Statements.
- The pLSI Property Statements in an ABEL-HDL source file override any defaults.

Specifying the Parameter File

A Parameter File is specified in ABEL-SUN by placing a “PLSI PROPERTY PARAM_FILE *file_name”* statement in your source file.

ABEL-SUN Design Example

The steps used to take a design created in ABEL-SUN from design entry through device programming are the same as those for a design created using ABEL-DOS. See “ABEL Design Example” on page 60 for an example.

Chapter 5 *Simulation*

You can simulate Synario designs using a Verilog simulator to perform both functional and timing simulation. You can perform functional simulation on ABEL-WIN designs using the Data I/O Equations simulator. You can perform timing simulation on ABEL-DOS and ABEL-SUN designs using the Viewlogic PROSim/ViewSim simulator or the LMC simulator. This chapter contains information on the following topics:

- Functional Simulation
- Timing Simulation

Functional Simulation

Synario Designs

The Synario simulator allows you to perform functional and timing simulation. You can either simulate your entire design, or just one of the sub-functions of your design, by creating a Verilog test fixture and associating it with the module you wish to simulate. The Synario simulator allows you to perform functional simulation at the project level or on a module-by-module basis.

To perform functional simulation:

1. Create a test fixture file for your design by using either the Synario text editor (as described in step 1.a) or an external text editor (as described in step 1.b).
   a. Select **Source** ⇒ **New** ⇒ **Verilog Test Fixture** from the Synario menu and click **OK**. The Associate Verilog Test Fixture dialog box appears. Click **OK**. Enter the name of your test fixture file (*design_name*.tf) in the New File dialog box and click **OK**. The Synario Text Editor window appears. Enter your test fixtures in this file and click **Close**. (See the DATA I/O *Verilog Simulation User Manual* for information on creating your test fixtures.) The *design_name*.tf file appears under the **Sources in Project** section of the Synario Project Navigator window.
   b. Select **Source** ⇒ **Import** and your simulation test fixture file (created with an external editor), *design_name*.tf appears under the **Sources in Project** section of the Synario Project Navigator window.
2. Click on ***design_name*.tf** file.
3. Double-click on **Functional Simulation**.
4. Click **Run** to start the Verilog simulator.

For more information on functional simulation, see the Data/IO *Verilog Simulator User Manual Synario Universal FPGA Design System*.

ABEL-WIN Designs

See the Data I/O *Equation and JEDEC Simulators User Manual Synario and ABEL* for information on performing functional simulation on ABEL-WIN designs.

ABEL-DOS/ABEL-SUN Designs

Pre-routed, functional simulation can be run in the ABEL environment using the ABEL PLASim application on ABEL designs created using ABEL-DOS or ABEL-SUN. PLASim verifies that the ABEL PLA file is functionally correct based on the Boolean equations generated within ABEL. For more information on PLASim, see the *ABEL Design Software User Manual*.

Timing Simulation

Synario Designs

Timing simulation with Synario designs is a post-route option. The pDS+ Fitter generates the Verilog (*design_name*.vlo) and standard design format (SDF) (*design_name*.sdf) files for timing simulation according to instructions specified through the Synario Process Properties Editor menu or in your Parameter File (*file_name*.par).

NOTE If you do not set the **Generate Verilog Timing Model** option to **Yes** using the Synario Process Properties Editor, you must include the following statement in your Parameter File or the .vlo and .sdf files will not be generated for timing simulation.

output_form verilog

If your design contains a register, the global reset pin of your device, !XRESET, must be referenced as described in step 1 below. Although this pin does not appear in functional simulation, it is automatically created during timing simulation if the design contains one or more registers.

Simulation results can be viewed from the Synario simulator waveform viewer, and logic levels back-annotated onto the top-level schematic for those designs which have a top-level.

To perform timing simulation:

1. Add the following statement to your ***design_name*.tf** file.

   ```
   t.d.\!XRESET = 1;
   ```
2. Select ***design_name*.tf** file under the Synario Project Navigator window.
3. Double-click on **Timing Simulation**.
4. Select **File** ⇒ **Setup** in the Setup Simulator window, and click either **Minimum Delay** or **Maximum Delay**. Click **OK**.
5. Click **Run** under the Synario Simulator window to start the timing simulation.
6. Select **Synario Simulator** ⇒ **Window** ⇒ **Waveform Viewer** to view the simulation results (optional).

ABEL-WIN, ABEL-DOS, and ABEL-SUN Designs

Timing simulation can be performed on ABEL designs using the Viewlogic ViewSim/PROSim simulator. See the *Lattice Viewlogic Simulation Library Supplement* for information on performing Viewlogic simulation on ABEL designs.

Lattice Semiconductor-Specific Simulation Signals

!XRESET

If you use an ispLSI 1016 or ispLSI 2032 device with Y1_AS_RESET=ON, !XRESET is created as an external pin. This pin drives !RESET. When this condition exists, specify only one of the two (!XRESET or !RESET) for post-route simulation.

XTEST_OE

Set the XTEST_OE pin to high if you use an ispLSI/pLSI 3192 or ispLSI/pLSI 3256 device.

Index

Symbols
!XRESEST
  simluation 71
!XTEST_OE
  simulation 71

A
ABEL
  Attributes 43
  Dot extensions 39
  ISTYPE statement 43
  Pin statement 58, 63
ABEL-DOS SmartPart->Options
  Device 58
  Ignore Preassignments 58
  Optimize Fit for Area 58
  Optimize Fit for Speed 58
ABEL-HDL exceptions 43
ABEL-SUN SmartPart->Options
  Attempt Preassignments 63
  Device 63
  Ignore Preassignments 63
  Keep Preassignments 63
  Optimize Fit for Area 63
  Optimize Fit for Speed 64
ABEL-WIN device kit 27
Attributes
  external pins 32, 35
  nets 32, 34
  pins 32, 35
  placing
    nets, pins, symbols 32, 45
  symbols 32, 35

C
CARRY_PIN_DIRECTION 45, 47
CASE_SENSITIVE 45, 47
CLK 32, 34
Compiler
  controlling 60, 66
Compiler Control Options
  CARRY_PIN_DIRECTION 45, 47
  CASE_SENSITIVE 45, 47
  EFFORT 45, 47
  EXTENDED_ROUTE 45, 47
  IGNORE_FIXED_PIN 45, 47
  ISP 46, 49
  ISP_EXCEPT_Y2 46, 49
  MAX_GLB_IN 45, 47
  MAX_GLB_OUT 45, 47
  OUTPUT_FORM 45, 47
  PARAM_FILE 45, 47
  PART 45, 48
  PIN_FILE 45, 48
  PULLUP 46, 49
  SECURITY 46, 49
  STRATEGY 46, 48
  TIMING_FILE 46, 48
  USE_GLOBAL_RESET 46, 48
  Y1_AS_RESET 46, 49
Controlling the pDS+ Fitter
  ABEL-DOS menu options 60
  ABEL-SUN menu options 66
  ABEL-WIN menu options 55
  Parameter File 51, 55, 60, 66
  pLSI Property Statements 51, 55, 60, 66
  Synario menu options 51
CRIT 32, 35

D
Design Attributes
  CLK 32, 34
  CRIT 32, 35
  GROUP 32, 34
  LOCK 32, 35
  LXOR2 32, 35
  PRESERVE 32, 34
  PROTECT 32, 35
  PULLUP 32, 35
  REGTYPE 32, 35
  SAP/EAP, SCP/ECP, SNP/ENP 32, 34
  SLOWSLEW 32, 35
  Synario menu equivalents 33
Design flow
  ABEL-DOS to pDS+ Compiler 16
  ABEL-SUN to pDS+ Compiler 16
  ABEL-WIN to pDS+ Compiler 14
  Synario to pDS+ Compiler 12
Design flow steps
  ABEL-DOS to pDS+ Compiler 17

ABEL-SUN to pDS+ Compiler 17
ABEL-WIN to pDS+ Compiler 15
Synario to pDS+ Compiler 13
Device declaration statement 59, 64, 65
Device kit
ABEL-WIN 27
Synario 26

E
EFFORT 45, 47
Entering Compiler Control Options
ABEL-HDL 54
ABEL-HDL designs 54
schematics 50
Entering Design Attributes
ABEL-HDL designs 34
schematic designs 33
Environment setup
ABEL-DOS 28
ABEL-SUN 29
ABEL-WIN 27
Synario 26
EXTENDED_ROUTE 45, 47

F
Files
.ABV 18
.JED 18
.PAR 18
.SDF 18
.SIM 18
.TF 18
.TMV 18
.TT2 18
.VLO 18
.WIR 18
Functional simulation
ABEL-WIN (Equations simulator) 69
Synario (Verilog simulator) 68

G
GROUP 32, 34

H
Help
hotline 30

I
IGNORE_FIXED_PIN 45, 47
Installation requirements
ABEL-SUN 25
Synario, ABEL-WIN, ABEL-DOS 24
ISP 46, 49
ISP_EXCEPT_Y2 46, 49
ISTYPE statement 43

L
Limitations
hierarchical Synario schematic 36
LOCK 32, 35
LXOR2 32, 35

M
MAX_GLB_IN 45, 47
MAX_GLB_OUT 45, 47
Memory requirements
ABEL-DOS 28
ABEL-SUN 29

N
Net attributes
CLK 32, 34
GROUP 32, 34
PRESERVE 32, 34
SAP/EAP, SCP/ECP, SNP/ENP 32, 34

O
OUTPUT_FORM 45, 47

P
PARAM_FILE 45, 47
PART 45, 48
Pin attributes
CRIT 32, 35
LOCK 32, 35
PULLUP 32, 35
SLOWSLEW 32, 35
PIN_FILE 45, 48
pLSI Property Statements
ABEL-HDL designs 34
PRESERVE 32, 34
PROTECT 32, 35
PULLUP 32, 35, 46, 49

R
REGTYPE 32, 35

S
SAP/EAP, SCP/ECP, SNP/ENP 32, 34
SECURITY 46, 49
Simulation
  !XRESET 71
  XTEST_OE 71
SLOWSLEW 32, 35
Specifying an LSC device
  ABEL-DOS 59
  ABEL-SUN (4.3) 64
  ABEL-SUN (5.1) 65
  ABEL-WIN 55
  Synario 51
STRATEGY 46, 48
Symbol attributes
  LXOR2 32, 35
  PROTECT 32, 35
  REGTYPE 32, 35
Synario device kit 26
Syntax example
  Design Attribute 34
  pLSI Property Statement 34

T
Timing simulation
  ABEL-DOS, ABEL-SUN, ABEL-WIN 70
  Synario (Verilog) 70
  Viewlogic (ViewSim/PROSim
    simulator) 70
TIMING_FILE 46, 48

U
USE_GLOBAL_RESET 46, 48

Y
Y1_AS_RESET 46, 49