ISP Software Basics

Introduction

This section explains the basic design flow necessary to compile the logic design and download programming for Lattice ispLSI® devices. A generic design flow for ispLSI and pLSI® using the Lattice proprietary Windows pDS Software tools and third-party tool sets in combination with the pDS+™ Fitters is described. The next section overviews the options available for ISP downloading and programming. The third section overviews ispATE™ programming using major vendors' hardware platforms for Automated Test Equipment (ATE). Finally, the design flow and software overview for the Lattice ispGDS™ software compiler and download options are described.

ISP Design Flow

After the conceptual stage of the logic or system design, a software design entry tool for implementing the logic functions, state machine or schematic design must be selected.

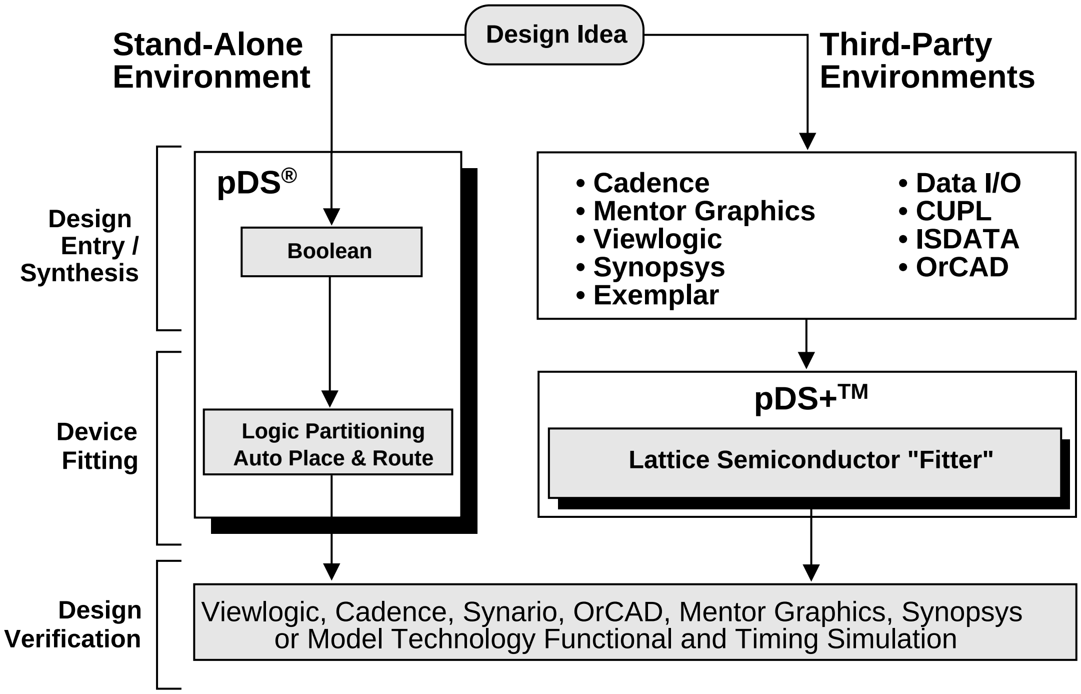
Lattice Semiconductor Corporation (LSC) supports proprietary and various third-party software tools which allow users a choice of design entry methods. These can include Boolean equation entry, state machine, hardware description language (such as VHDL) or schematic entry. The goal is to consolidate the logic functions into a reduced set of equations that can be compiled for a given device hardware platform.

The LSC proprietary pDS® Software allows the user to enter the design in Boolean equation format in the Windows environment. The LSC pDS+ Fitters interface to major third-party platforms, allowing designers to use familiar and existing tool sets. The pDS+ fitter converts intermediate files from third-party design tools directly to a standard JEDEC file for programming. The pDS+ software also allows the addition of attributes and parameters to be attached for control of fitting, partitioning and device-specific functional options such as pull-up resistors or slew rate control on a pin by pin basis. pDS+ Fitters can export the necessary simulation netlists for third-party simulators, allowing both functional and timing simulation for design verification.

pDS Software Summary

LSC offers a proprietary tool set called the pDS pLSI / ispLSI Development System. The Windows-based pDS software is developed and supported by LSC and offers a low-cost, high-performance, Windows-based text de-

Figure 1. pDS and pDS+ Design Flows



sign entry. The entry uses simple Boolean equations or macros that consist of a standard library and a TTL library. After a design has been entered, the software runs automated syntax checking for design syntax violations. Partitioning of the logic is under control of the designer, the placement and routing is automated by the software. After the design has passed these processes, the pDS tool generates the necessary JEDEC files for programming which can be downloaded within the pDS environment for ispLSI devices. For simulation, the pDS software tool exports static timing tables. It also exports simulation netlists for Viewlogic's Viewsim, ORCAD's VST and Data I/O's Synario simulation package.

pDS Software Features

- ispLSI AND pLSI DEVELOPMENT SYSTEM
  - — Supports ispLSI and pLSI 1000/E and 2000 Series Device Families
  - — Upgrade to Support ispLSI and pLSI 3000 and 6000 Device Families
- DESIGN ENTRY WITH EASY-TO-USE WINDOWS ENVIRONMENT
  - — ABEL-Like Boolean Equation Entry
  - — Logic Macro Entry with Over 275 "TTL-Like" Macros and Over 200 TTL Macros
  - — Manual Device Partitioning Ensures Tight Control of Performance and Utilization
- FAST DESIGN COMPILATION
  - — Multi-Level Logic Synthesis
  - — Design Optimization and Minimization
  - — "Hands-Free" Automatic Place and Route
  - — Predictable Performance
- COMPLETE DESIGN VERIFICATION
  - — Functional and Timing Simulation Options
- INDUSTRY STANDARD JEDEC PROGRAMMING FILE GENERATION
  - — Standard JEDEC Fuse Map
- IN-SYSTEM PROGRAMMING SUPPORT
  - — ispCODE C Source Routines Included
  - — ISP™ Daisy Chain Download
  - — ispATE Board Test Programming Utility

PLATFORMS SUPPORTED
- — PC Windows 3.1 / Windows 95 / Windows NT

pDS software is a complete entry package that allows designers high utilization of logic device resources. The easy-to-use Windows interface with Boolean expression text entry along with the System Macro Library and TTL Macro Library offer a low-cost opportunity to design with Lattice Semiconductor's ispLSI and pLSI devices.

pDS Software also contains built-in ISP Download functions to program multiple ISP devices in a serial daisy chain.

For further details on pDS software, see the pDS Software and isp Starter Kit data sheets.

pDS+ Fitter Summary

pDS+ software packages are the interface to major third-party vendors' entry packages. The pDS+ software (or Fitter) combined with third-party entry tools offer the advantages of high-end entry methods with a common tool set, and the advantages of interfacing with a previously installed software tool database. This combination provides a complete software solution—the third-party tool used to implement the design entry and verification along with the Lattice pDS+ Fitter for partitioning and routing of internal logic.

Lattice Semiconductor has teamed with the software vendors listed in Table 1.

ispGDS Compiler Support

To simplify the development of ispGDS devices, Lattice offers an ispGDS assembler named "GASM" which processes the input ASCII files to generate the JEDEC compatible fuse map files required for the ispGDS devices. Free ispGDS assembler software is available from the Lattice Hillsboro BBS at **503-693-0215** under the **GDSPKG.ZIP** file. This software is also available on diskette by calling the Lattice Hotline at **1-888-ISP-PLDS**. For design engineers familiar with standard third-party compiler software packages, ABEL from Data I/O and CUPL from Logical Devices also support all ispGDS devices.

Using the ispGDS Compiler

The compiler will accept an ASCII text file containing the ispGDS programming instructions, and will create JEDEC and .DOC files.

Compiler Syntax

The basic compiler syntax supports inserting comments, title, device type, pin assignments and input/output assignments. The ispGDS compiler source file comment lines are denoted with quotation marks at the beginning of the comment lines. The title is defined with the key

Table 1. Third-Party Software Vendors

Vendor	Design Tool	For More Information, See...
Cadence	Concept, Synergy, Verilog-XL, Leapfrog	1996 Lattice Semiconductor Data Book, pDS+ Cadence Software Data Sheet
Data I/O	ABEL, Synario	1996 Lattice Semiconductor Data Book, pDS+ ABEL and pDS+ Synario Software Data Sheets and ISP Synario System Data Sheet
Exemplar Logic	Galileo Logic Explorer	1996 Lattice Semiconductor Data Book, pDS+ Exemplar Software Data Sheet
ISDATA	LOG/iC Classic, LOG/iC 2	1996 Lattice Semiconductor Data Book, pDS+ LOG/iC Software Data Sheet
Mentor Graphcs	Design Architect, Autologic, Autologic II, Quicksim II, Quick VHDL, System V (Model Technology)	1996 Lattice Semiconductor Data Book, pDS+ Mentor Software Data Sheet
Logical Devices	CUPL	1996 Lattice Semiconductor Data Book, pDS+ CUPL Software Data Sheet
OrCAD	SDT, PLD, VST 386+, Capture for Windows, Simulate for Windows	1996 Lattice Semiconductor Data Book, pDS+ OrCAD Software Data Sheet
Synopsys	Design Compiler Expert and Professional, FPGA Compiler, VSS	1996 Lattice Semiconductor Data Book, pDS+ Synopsys Software Data Sheet
Viewlogic	PRO Series, WorkView PLUS, Powerview, WorkView Office	1996 Lattice Semiconductor Data Book, pDS+ Viewlogic Software Data Sheet

word “title = .” Any text following the “title =” key word that is within single quotes is defined to be the title of the design. Similarly, the device type is defined by the key word “device =” followed by one of the three valid device types – ispGDS22, ispGDS18, ispGDS14. The compiler syntax also allows the user to assign pin names by typing in a 10 character pin name followed by at least a single space, the “pin” key word and the pin number. This pin assignment is optional since the compiler syntax allows the user to use the “pin” key word and the pin number directly in the input/output assignments.

The output pins are assigned on the left side of the equation and the input pins are assigned on the right side of the equation. To assign an output pin to either high or low, simply assign “H” or “L” respectively on the right side of the equation. If you need to assign an input pin to multiple output pins, use one line for each assignment, as shown in the following example. In the example below, pin 28 is an input that is routed to three outputs — pin 1, pin 2 and pin 3. Furthermore, each output’s polarity can be individually defined. The example shows pin 3 as an active low polarity whereas pin 1 and pin 2 are defined to be active high polarity.

```
 pin 1 = pin 28
 pin 2 = pin 28
!pin 3 = pin 28
```

Assembling a File

To use the assembler, create an ASCII ispGDS source file, then invoke the assembler from the DOS command line. For example:

```
gasm <test.gds>
```

where test.gds is the name of the ispGDS source file. GASM will create a JEDEC file with the same base name, and a .JED extension, like “test.jed,” and a doc file with a .DOC extension, like “test.doc.” Listing 1 illustrates an ispGDS source format.

Device Programming Options

Lattice Semiconductor has pioneered the E^{2}CMOS Programmable Logic Device (PLD) industry and is the inventor of In-System Programmability™ (ISP). In-System Programmability allows the flexibility of programming devices that are soldered to the board. Changes to logic functions within the device can be made in-system, enhancing the prototyping and manufacturing stages as well as end user upgradability.

There are several options for programming Lattice Semiconductor’s ispLSI/pLSI, ispGAL®/GAL® and ispGDS™ devices. Conventional methods use third-party hardware programmers to download the standard JEDEC file to the target device (see the *Third-Party Programmer* document). All Lattice products can be programmed using qualified hardware programmers.

The second method of programming involves ISP or In-System Programming. This can be done with all of LSC’s ispLSI, ispGAL and ispGDS products. ISP offers the user several options for ISP downloading, the easiest of which is ISP Daisy Chain Download Software. Other options include C-Code for custom front end or menu interfaces, Embedded System or microprocessor programming, remote programming and ispATE, programming on an automated tester.

All of these methods use similar algorithms available in a compiled format or as uncompiled code for user modification. The heart of ISP download programming is the Lattice ISP state machine as described in the *Hardware Basics* document. The state machine is built into each ISP device and allows 5-Volt In-System Programming using TTL-level signals to shift a serial ISP stream into the device or serial chain of ISP devices.

Listing 1. ispGDS Source Format

The following text is an example of a ispGDS source file.

```
"This is a comment (line begins with quotation mark)
title = 'DIP SWITCH REPLACEMENT CONFIGURATION'

" the ispgds device type (ispgds22, ispgds18, ispgds14)
device = ispgds22

" pin names are defined as follows

pin_name  pin 28

" pin 1 is an output connected to pin 28
pin 1 = pin_name
pin 2 = pin 27

" pin 3 is another output connected to pin 28

pin 3 = pin 28

" pin 5 is always high
pin 5 = h

"pin 6 is always low
pin 6 = l
pin 8 = pin 22

"! defines the inverted output for pin 9
!pin 9 = pin 20

pin 10 = pin 19
pin 12 = pin 17
pin 13 = pin 16
pin 14 = pin 15
```

Programming Option Summary

ISP Daisy Chain Download Software from Lattice supports the programming of all Lattice ISP devices in a serial daisy chain programming configuration in a PC environment. The software is supported in Windows (ISP Daisy Chain Download for Windows) and DOS (ISP Daisy Chain Download for DOS). The ISP programming interface is driven from the parallel port of the PC and requires an external ispDOWNLOAD™ Cable for programming single devices or multiple devices in a serial daisy chain. A more thorough description of ISP Daisy Chain Download software is included in the *In-System Programming on a PC* document.

ispCODE Software is used for programming ispLSI, ispGAL and ispGDS devices. It consists of C source code routines and programming functions. The modular code routines and functions can be easily modified to a customer's needs to produce dedicated ISP programming software interfaces. The C source code can be ported to different applications and platforms. The routines can be compiled as is to produce a Windows interface that ports the ISP programming signals to the parallel port of a PC.

After the design for the logic functions have been compiled into a JEDEC file from a design entry package, the JEDEC file is converted to an ispSTREAM™. This stream format is the binary equivalent of the JEDEC file. The ispSTREAM is used when the device is programmed and verified. It contains all the JEDEC information in a compressed format.

ispATE Software converts a single JEDEC file or a group of JEDEC files to a set of programming vectors for automated test equipment. The vector set is then used to download from the tester to the target devices soldered on the board. This enhances the manufacturing flow and eliminates stand-alone programming steps during production or assembly. ispATE software is Windows- based and supports the following tester platforms along with a generic programming vector set: the Teradyne Z1800 series, the GenRad GR228X series and Hewlett-Packard's HP3065 and HP3070 families of board testers

All ISP programming is done with TTL-level signals and no high voltage is necessary to drive the programming signals. Five ISP programming signals are needed from the tester. These can be interfaced using standard test probes or any generic pin interface. Programming at board test offers the advantage of minimized device handling and requires no dedicated hardware programmer. Testability is also a key advantage. Test patterns can be programmed into ISP devices to exercise various portions or functions within the system that may not be testable or observable.

A second option for ATE programming is to build a test program based on a higher-level programming language such as C. By using a high-level language, the test time can be improved due to the number of test vectors involved. Lattice Semiconductor supports programming utilities based on C. ispCODE™ is a set of C source code routines used to program a single device or a chain of ISP devices. The code is modular and can be customized to any given application and front end interface.

Refer to the *ATE Programming of ISP Devices* document and the ispATE Software data sheet for additional information.

ISP Programming Using an Embedded Processor

Another option that broadens the application of ISP devices is the use of an embedded microprocessor to program the devices. Since the programming interface uses only five wires and its signals are TTL-level, it is easy to adapt ISP programming algorithms and signals for use on a microprocessor or embedded microcontroller. There are two ways this can be accomplished: the microprocessor can act as an ISP controller, receiving its commands from a host processor or the stand alone processor can contain the micro-code and possibly the JEDEC file data or ispSTREAM file. The code necessary to program and control the ISP programming state machine is small enough to be stored as program memory on an EEPROM type microcontroller.

Applications for embedded programming are very broad and may include remote re-programming for update purposes or multiple function PC cards that can be self-configured by a combination of software and firmware. See the *In-System Programming from an Embedded Processor* document for further information.

Lattice Semiconductor Corporation (LSC) products are made under one or more of the following U.S. and international patents: 4,761,768 US, 4,766,569 US, 4,833,646 US, 4,852,044 US, 4,855,954 US, 4,879,688 US, 4,887,239 US, 4,896,296 US, 5,130,574 US, 5,138,198 US, 5,162,679 US, 5,191,243 US, 5,204,556 US, 5,231,315 US, 5,231,316 US, 5,237,218 US, 5,245,226 US, 5,251,169 US, 5,272,666 US, 5,281,906 US, 5,295,095 US, 5,329,179 US, 5,331,590 US, 5,336,951 US, 5,353,246 US, 5,357,156 US, 5,359,573 US, 5,394,033 US, 5,394,037 US, 5,404,055 US, 5,418,390 US, 5,493,205 US, 0194091 EP, 0196771B1 EP, 0267271 EP, 0196771 UK, 0194091 GB, 0196771 WG, P3686070.0-08 WG. LSC does not represent that products described herein are free from patent infringement or from any third-party right.

The specifications and information herein are subject to change without notice. Lattice Semiconductor Corporation (LSC) reserves the right to discontinue any product or service without notice and assumes no obligation to correct any errors contained herein or to advise any user of this document of any correction if such be made. LSC recommends its customers obtain the latest version of the relevant information to establish, before ordering, that the information being relied upon is current.

LSC warrants performance of its products to current and applicable specifications in accordance with LSC's standard warranty. Testing and other quality control procedures are performed to the extent LSC deems necessary. Specific testing of all parameters of each product is not necessarily performed, unless mandated by government requirements.

LSC assumes no liability for applications assistance, customer's product design, software performance, or infringements of patents or services arising from the use of the products and services described herein.

LSC products are not authorized for use in life-support applications, devices or systems. Inclusion of LSC products in such applications is prohibited.

LATTICE SEMICONDUCTOR CORPORATION
5555 Northeast Moore Court
Hillsboro, Oregon 97124 U.S.A.
Tel.: (503) 681-0118
FAX: (503) 681-3037
http://www.latticesemi.com

November 1996