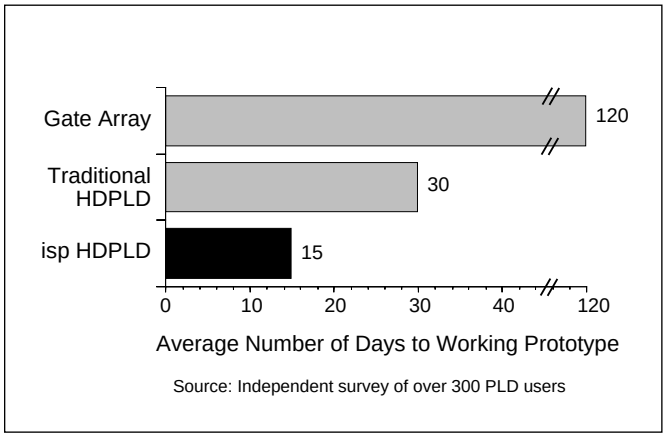
ISP Overview

Introduction

ISP™ (In-System Programmability): The ability to reconfigure the logic and functionality of a device, board, or complete electronic system before, during, and after its manufacture and shipment to the end user.

ISP is the new standard in programmable device technology. ISP eliminates traditional PLD limitations and delivers benefits in board and system-level design, manufacturing, and programming. Since ISP hardware is as flexible and easy to modify as software, design upgrades are simple. Because ISP devices can be treated like any other device on the PCB, no special manufacturing flow is required to program ISP devices; standard 5-volt logic level programming signals are easily generated by a PC, Sun Workstation®, ATE (Automatic Test Equipment), or system embedded microprocessor. In pioneering ISP, Lattice has developed an integrated solution of silicon, software and applications know-how that makes ISP a practical technology.

Figure 1. In-System Programmability: Time-To-Market Advantage



Q: What's Driving ISP Momentum?

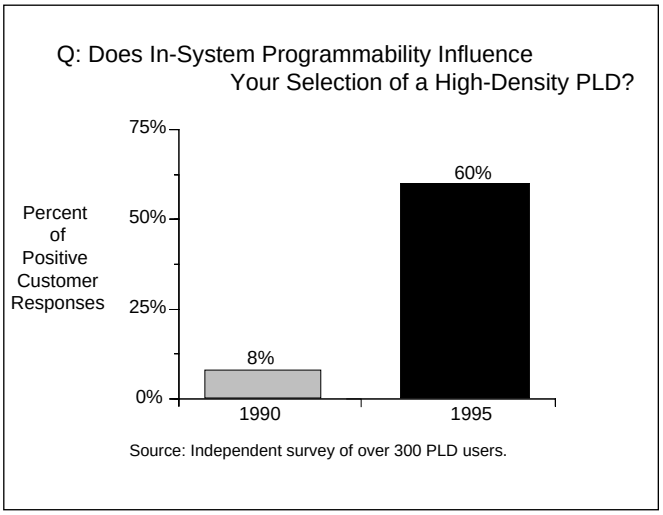
A: Time-To-Market

The drive for a shorter time-to-market has fueled explosive growth in the use of PLDs.

Based on user responses, ISP provides an additional 50% reduction in time-to-market over traditional HDPLDs and a more than 85% reduction in time-to-market compared to gate array implementations (Figure 1).

Another indicator of ISP's momentum is the percentage of designers who say that ISP capability will influence their selection of an HDPLD (Figure 2). Just five years ago, when asked, only 8% of system designers said that ISP would influence their HDPLD decision. Today, that percentage has leaped to 60%!

Figure 2. In-System Programmability: An Emerging Standard



This overview presents the benefits of ISP and summarizes the ISP solutions available from Lattice. The outcome is convincing—ISP drives dramatic savings in design cycle time, manufacturing costs, and time-to-market.

ISP Design Benefits

ISP allows design, test, and manufacturing engineers to reconfigure system features while the devices remain soldered on the circuit board. This capability revolutionizes design prototyping, board-level debug, system manufacturing, and system upgrades.

The Superior Prototyping Solution

During most system design cycles, major board building blocks such as the microprocessor and RAM are selected first, well before system logic decisions are made. When using ispLSI devices, the designer can fully populate his prototype board with the major building blocks, interconnecting all functions with programmable logic and switch devices. Design changes, whether they require added or modified logic, can be made in minutes using Lattice's pDS or pDS+ software design tools. A 5-wire download cable from a PC or workstation to the prototype board downloads the new logic into the

device(s). This ability to modify system functionality without changing components or printed circuit board (PCB) layout is only the first of many advantages afforded by Lattice's ISP technology.

Internal Test

Once the ISP logic has been stabilized, the designer may use the ISP devices to debug other portions of the board. For example, a circuit board frequently operates in a system where it is supplied with stimulus from other boards. The designer can use in-system programmability to debug system-level operation more quickly by reconfiguring the ISP devices to force or redirect signals (e.g. clocks or control signals) into various portions of the board design. This ability to thoroughly check board designs saves precious time during system-level debug and translates directly into a competitive time-to-market advantage.

Board Reconfiguration and Field Upgrades

ISP devices provide an ideal way to reconfigure boards and/or upgrade product features in the field. With conventional logic technology, a system installed at a customer site is very expensive and difficult to upgrade to the latest hardware revision, to fix hardware bugs, or to enable hardware options. With ISP devices, however, if a subsequent reconfiguration, upgrade or repair is required, a simple upgrade disk can be used, either in the field or the factory, to reconfigure the logic (e.g. to modify memory refresh or control logic or to operate with a faster microprocessor). Updates via modem, serial link, or a special ISP programming interface are possible depending on the system environment or needs.

ISP Manufacturing Benefits

ISP is not only revolutionizing the world of logic design but is also dramatically transforming the world of manufacturing. The ISP devices support multi-function hardware designs that reduce system part count and cost. ISP also supports reconfigurability for test which enhances board-level testability and, ultimately, system reliability. Finally, ISP allows the “standard PLD manufacturing flow” to be simplified (Figure 3), reducing cost and enhancing system quality.

Multi-Function Hardware

ISP can be used to exploit the concept of multi-function hardware: a single hardware design able to implement a variety of system-level functions via in-system programming. Multi-function hardware allows manufacturers to reduce the number of unique board designs used in a system, further simplifying the manufacturing flow.

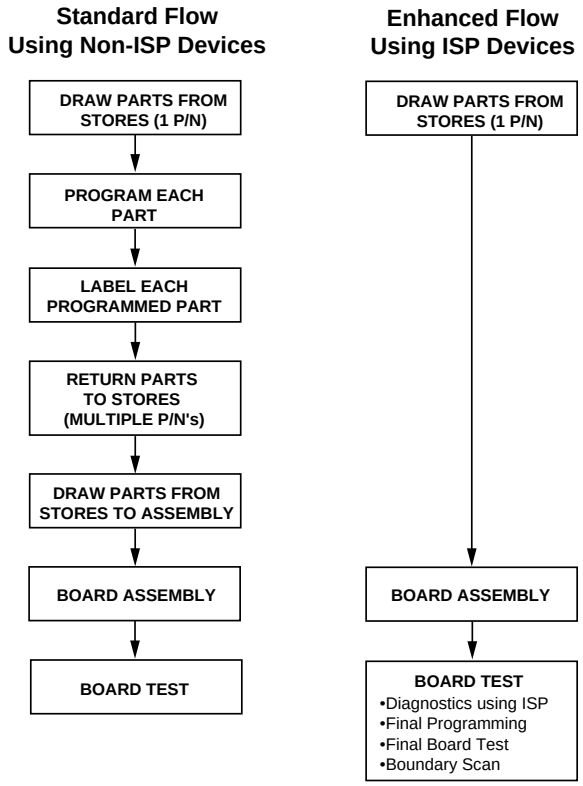
Multi-function hardware dramatically lowers system-level costs by reducing the component count on the boards as well as reducing the number of different boards required to implement various system-level options.

A dual-processor board, intended to interface with several bus interface standards, illustrates these benefits. The traditional solution calls for dedicated logic for each of the bus interface standards, requiring either a unique board dedicated for each standard or a single board with additional logic. ISP devices allow the design of a single generic bus interface, which can be configured in-system to interface with each of the bus standards, saving components, and cost.

Reconfigurability for Test

The ISP approach facilitates board-level testing and increases system fault coverage without sacrificing board resources or real estate. A diagnostic test pattern can be temporarily programmed into the ISP devices to exhaustively exercise board-level functions. Additionally, with ispGDS, programmable signal routing can be exploited in the test environment to perform enhanced board-level

Figure 3. ISP Manufacturing Flow vs. Standard Manufacturing Flow



test. For example, certain ispLSI devices may be configured by the tester to force test sequences into other portions of the board logic. The tester then monitors the response of this action and determines if the board passes or fails. This ability to detect board-level failures early in the manufacturing cycle reduces overall system cost. Once these detailed diagnostics are complete, the ISP devices can be reprogrammed to their normal logic configurations for final functional testing.

Boundary Scan

Complementing the ISP approach to board-level testing, IEEE Standard 1149.1 Boundary Scan technology (available with the 3000 and 6000 series) enhances overall system quality. As component densities on the system boards increase, along with greater chip density and I/O, the ability to access and test critical nodes is impaired. With Boundary Scan Test, a serial interface through the test access port (TAP) simplifies field diagnostics and testing while costs are reduced. And because the same Boundary Scan serial path and control pins are used for implementing ISP programming, overall manufacturing costs are reduced as well.

Simplified Manufacturing Flow / No Bent Leads

At present, there are no automatic handlers capable of handling the programming of high lead-count, high-density Quad Flat Pack PLDs. As a result, all non-ISP high lead-count devices must be programmed by hand using a standard logic programmer.

It is a difficult task to insert a high lead-count, small lead-pitch device into a programming socket adapter, program, label (or mark) and reinventory the device without bending the delicate package leads. These bent leads can result in poor coplanarity and bad solder connections, increasing the amount of board and system-level troubleshooting required.

With the ISP devices, the parts go directly from the receiving dock to the manufacturing floor for placement on the PCB, entirely eliminating the stand-alone programming and mark operations and avoiding bent leads associated with misalignment of the device in the programmer socket. Unprogrammed ISP devices can be loaded into auto-insertion equipment and then placed directly onto the PCB without sockets or regard for the specific logic configurations. Individual device configurations can be downloaded from Automatic Test Equipment, PC, or workstation platforms at final board test. Programming of high-density PLDs containing thousands of gates takes only seconds.

System Upgrades and Repair

Lasting benefits from the use of ISP can be realized even after systems are shipped. In-system reprogramming can reduce field maintenance costs through enhanced field diagnostic capability, less costly product feature upgrades, and simpler maintenance procedures. Training, documentation, and on-going support can also be simplified by using the ISP approach to build in maintainability.

ISP Applications

Lattice's breadth of ISP device options, together with their leading-edge performance and features, have resulted in the design-in of ISP devices into a wide range of electronic systems. These applications include:

- Multimedia Video Editing
- Electronic Test Equipment
- Network Routers and Bridges
- Cellular Telephone Base Stations
- Telephone Switching Systems
- Hardware Accelerators
- Memory Subsystems
- Multi-Standard Video Frame Grabber
- Data Acquisition
- Image Processing

Why have designers embraced the ISP concept? For many, the manufacturing cost benefits, faster logic design and prototyping, and ability to reliably program high-pin count devices have been the most obvious benefits of employing ISP. However, in addition, ISP's ability to reconfigure systems immediately prior to and after shipment has begun to open up new possibilities.

For example, a very common but practical application for ISP comes from a company manufacturing traffic signal controllers. These controllers support priority "green lights" for emergency vehicles and buses through strobe light sensors that detect coded strobe sequences from the vehicles. The authorized sequences vary from city to city. ISP allows the sequence detector to be reprogrammed easily at the time the signal controller is shipped to a particular area or after it is installed. The alternative of custom-coded, traditional PLDs would result in significant additional effort and expense to customize the hardware of each system.

ISP Overview

E²CMOS Technology

Lattice was the pioneer in electrically erasable CMOS (E^2CMOS) technology with the invention of the GAL device. With over 500 million devices shipped to customers, Lattice has the most CMOS PLD manufacturing experience in the industry. Merging E^2CMOS technology with ISP, Lattice conceived the in-system programmable, electrically erasable CMOS process. All ispLSI, ispGAL, and ispGDS devices are manufactured using Lattice's proprietary high-speed UltraMOS E^2CMOS technology. Lattice is unique among "fab-less" companies in that it performs its own process technology development. UltraMOS technology successfully combines the best features of CMOS and NMOS process technology to yield PLDs with the following key features:

- Industry Leading Performance
- High Logic Densities
- Low Power Consumption
- Fast Erase and Reprogram Times (Seconds)
- 100% Full Parametric Testability
- 100% Programming and Functional Yields

Lattice's experience in E^2CMOS manufacturing allows it to specify the best ISP parameters in the industry, including ISP programming over the full Commercial temperature range (0˚ to 70˚ C), a minimum of 10,000 program / erase cycles for ispLSI, ispGAL and ispGDS devices and 20-year guaranteed program retention.

Unlike SRAM-based programmable devices, the non-volatility of E^2CMOS means there is never a need to reprogram ISP devices after a power-down and power-up sequence. In addition, there is no need for a separate memory component to store the logic program. A "security cell" feature is also available, allowing the device to be programmed, verified, and then secured. After the device has been secured, the binary pattern cannot be read from the device. However, even a secured device can be identified via the User Electronic Signature (UES). This field is reserved for the user to record product data such as code revisions and device functions.

ISP Solutions from Lattice

Lattice offers eight families of In-System Programmable devices (Table 1).

The ispLSI® Families of High-Density Programmable Logic from Lattice

The ispLSI families merge ISP technology with Lattice's high-performance, high-density pLSI® (programmable Large Scale Integration) architecture. The ispLSI devices are the first in-system programmable logic devices to combine the performance and ease of use of PLDs with the density and flexibility of FPGAs. Their powerful architecture can implement a wide range of logic functions including registers, counters, multiplexers, decoders, and complex state machines. With 154MHz system speed and logic densities ranging from 1,000 to 25,000 gates, they're the most powerful programmable logic components available today.

Table 1. ISP Solutions from Lattice

	The Premier High-Density PLD 1K, 1KE, 2K, 2KLV, 3K & 6K Families	World's First In-System Programmable 22V10	In-System Programmable Switch Matrix
Density	1-25K Gates	500 Gates	7x7-11x11 Matrix
Speed: t_{pd} (ns)	5.5	7.5	7.5
Speed: f_{max} (MHz)	154	111	50
Macrocells	32-320	10	N/A
Registers	32-512	10	N/A
Inputs & I/Os	34-258	22	14-22
Packages	44-, 68-, 84-PLCC 44-, 48-, 100-, 128-, 176-TQFP 120-, 128-PQFP 160-, 208-, 240-, 304-MQUAD	28-PLCC	20-, 24-, 28-PDIP 20-, 28-PLCC

The ispGAL® Family of Low-Density Programmable Logic from Lattice

The ispGAL family brings ISP technology to Lattice's industry standard GAL® (Generic Array Logic) family of Low-Density PLDs. The ispGAL22V10 combines the full functionality of the popular GAL22V10 architecture together with ISP technology, while maintaining the GAL22V10 standard 28-pin PLCC package and footprint.

At 7.5ns Tpd and 111MHz system speed, the ispGAL22V10 is ideally suited for high-speed, small- to medium-scale logic functions typically found at the heart of today's microprocessor based systems.

The ispGDS™ Family of Programmable Digital Switches from Lattice

The ispGDS (in-system programmable Generic Digital Switch) family represents the expansion of ISP technology beyond system logic to system interconnect. This merger of ISP and a switch matrix architecture provides the ability to quickly implement and change p.c. board connections without changing mechanical switches or other system hardware. These high-performance, low-power, programmable digital switch devices are offered in a variety of matrix sizes and packages adding system flexibility to users.

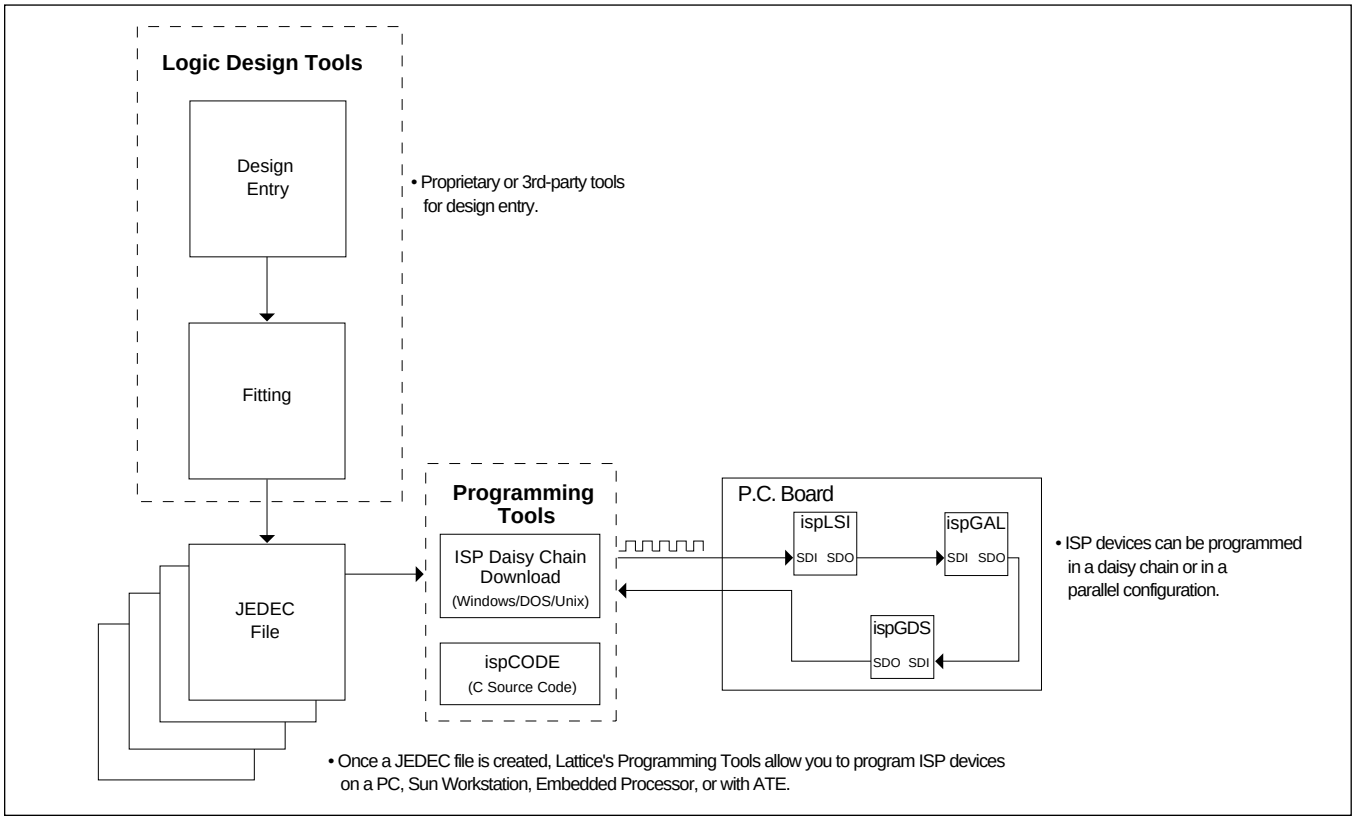
The ispGDS family is an ideal solution for easy, end-system feature configuration. With its 7.5ns performance, the ispGDS family also supports high-performance signal routing applications. The result is system hardware that can be reconfigured under software control without manual intervention.

ISP Implementation

ISP Interface

Programming and reprogramming ISP devices is simple and straightforward (Figure 4), requiring only a 3.3 or 5-volt power supply and a simple 4- or 5-wire serial interface depending on which ISP device is used. In-system programming operations such as PROGRAM, VERIFY, and ERASE are performed by passing commands and data to the ISP devices over the serial in-system programming interface (Figure 5). The basic programming signals consist of Serial Data In (SDI), MODE select (MODE), Serial Data Out (SDO) and Serial Clock (SCLK). In addition, $\overline{\text{ispEN}}$ is used to enable or disable the other four programming control signals on ispLSI devices, allowing these four pins to also function as dedicated inputs during normal operation.

Figure 4. ISP Design and Implementation Flow



Device Programming Configurations

Single or multiple In-System Programmable (ISP) devices can be programmed in several configurations. Each ISP device can be programmed individually, through an independent ISP interface, or multiple devices can share a parallel multiplexed or serial daisy chained interface. The serial daisy chain configuration, shown in Figure 5, is the most efficient and easiest to implement, as it utilizes a simple hardware interface and programming procedures.

Multiple Programming Platforms

Lattice's ISP devices can be easily programmed in a wide variety of ways:

PC

With ISP, engineers save time and money by designing and programming logic on a single platform. Designers can enter and simulate designs using popular third-party CAE tools, automatically place-and-route the logic using Lattice's pDS or pDS+ tools, and then download the programming files to ISP devices without leaving their seat. With ISP, device programming is easy; Lattice's ispDOWNLOAD™ cable connects to the PC parallel port to give quick, easy and inexpensive programming of one or more ISP devices.

ATE

Lattice's ISP devices can also be programmed at final board test on an ATE, completely eliminating the need for a third-party device programmer for production. This streamlines the manufacturing flow and allows programmable devices for the first time to be treated like any other components on the board. To make the task easy, Lattice's ispCODE™ and ispATE™ tester programming utilities can be used to generate programming test vectors, or programming routines based on ispCODE can be written in the tester's high-level language.

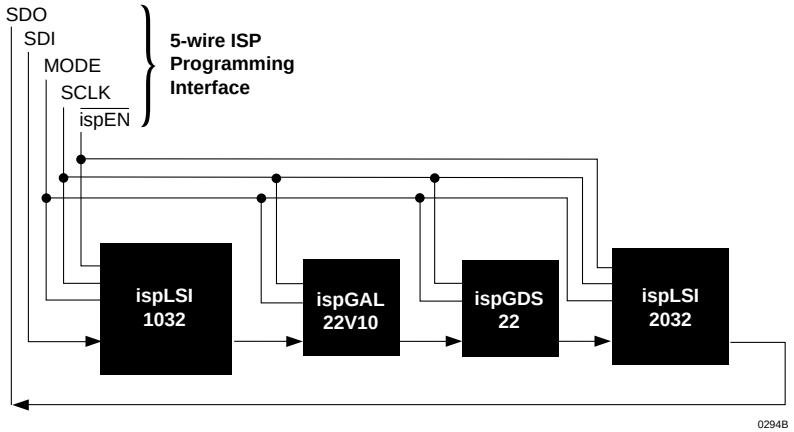
Embedded Processor

System Designers who want to be able to modify product features or upgrade their system hardware after the product has shipped to their customers will be interested in in-system programming using their product's own embedded processor. The product's embedded processor can be used to directly supply the ISP programming signals through a simple 4- or 5-bit port. Logic fuse maps and code can be stored in EPROM or other available system memory element. If the system has a modem or network link, remote download of new configurations from a central point is even possible. As result, systems no longer become obsolete as soon as they leave the factory, but can adapt and change to meet customers' growing needs for years to come.

Third-Party Programmer

Finally, ISP devices can be programmed by a number of popular third-party programmers. In some cases, a user owns a third-party device programmer and is interested in the long-term benefits of ISP (like future field upgradability) but isn't ready to make the change yet. These users can still program ISP devices like any other PLD on their device programmer. If they do need to change the device programming pattern at some future date, ISP will certainly make the process a lot easier: no device desoldering, no board swapping, no wasted effort.

Figure 5. In-System Programming Interface



Development Tools

Lattice ispLSI families are supported by a host of fully integrated development tools from Lattice and leading third-party CAE tools vendors. These easy-to-use, powerful tools fully support the three primary steps in the Lattice ispLSI design flow: design entry, device fitting, and design verification. From the power of VHDL synthesis to the simple and complete ispStarter Kit, Lattice has the tools you need for ISP design.

pDS and the ispStarter Kit

Lattice's pDS software quickly and easily takes your design ideas from design entry to programming. This Windows-based Boolean logic data entry tool supports over 225 macros, 150 TTL functions, and user-definable macros to speed the design process.

pDS automatically verifies the logic design, performs logic minimization and checks for signal availability. The Lattice Place and Route module allows you to assign pins and critical speed paths to ensure highest performance and utilization.

Lattice's ispStarter Kit is a version of the pDS tool supporting 44-pin ispLSI device design. This inexpensive, yet complete system contains everything you'll need to begin designing with Lattice ISP: software, documentation and hardware, including ispLSI, ispGAL and ispGDS samples as well as ispDOWNLOAD cables.

Lattice Third-Party Tool Philosophy

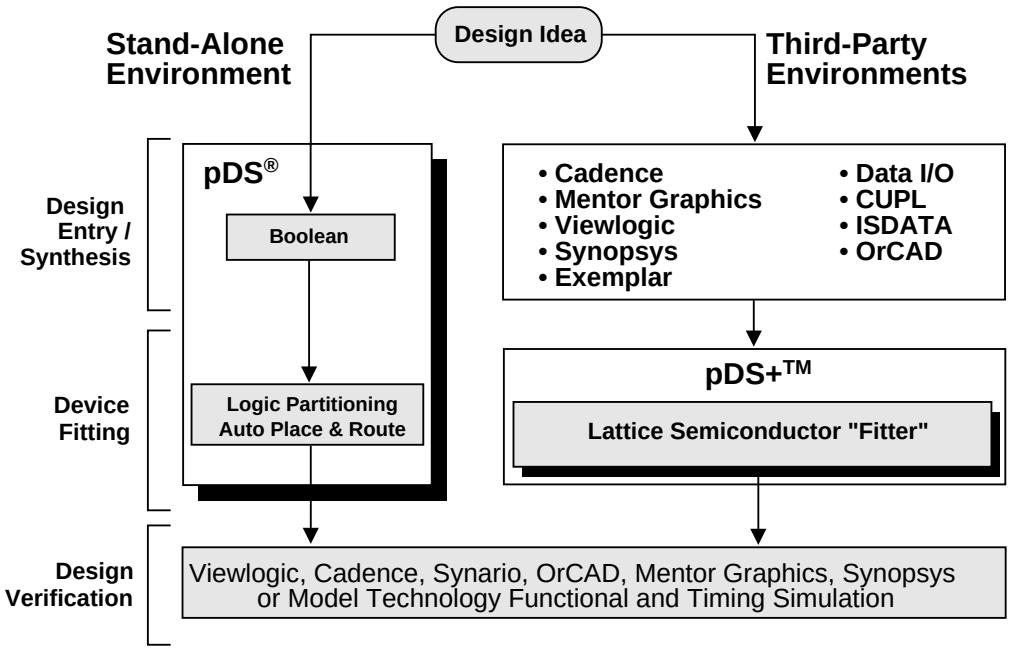
Whether you choose the ispStarter Kit or a highly integrated CAE vendor design environment, Lattice supports your needs. Since the introduction of our ispLSI technology, we've been working closely with the most popular CAE tools vendors to provide design solutions. Today Lattice offers solutions supporting all the most popular CAE vendors' design environments. The pDS+ Fitter is tightly integrated with the likes of Cadence, Data I/O, Mentor Graphics, OrCAD, Synopsys, Viewlogic and others. The Lattice third-party CAE vendor support offering is one of the most robust in the industry and will continue to grow. Couple these powerful, familiar schematic, synthesis, and verification tools with the pDS+ Fitter and ISP becomes easier than you ever expected.

pDS+ and CAE Vendor Interfaces

The Lattice pDS+ Fitter is designed to be tightly integrated with third-party CAE vendor design environments. Schematic capture, state machine and VHDL entry are supported through an expanded macro library (>300 macros). Getting started is easy because the design environment is the familiar one you always use.

Once a design is complete, a netlist is imported into the Lattice pDS+ Fitter for implementation. The pDS+ Fitter uses architecture-specific algorithms to synthesize a logic description for the ispLSI device. It then provides

Figure 6. Development Tools Available from Lattice



multi-level logic synthesis, automatic partitioning, and automatic place and route to maximize device utilization and performance.

The pDS+ Fitter also works with third-party CAE tools to support design verification. Functional and timing libraries support various third-party simulators. Following design verification, the pDS+ Fitter generates a JEDEC fusemap for Lattice ISP device programming.

Lattice's CAE tool solutions give designers powerful, easy-to-use, cost effective design tools to meet their development needs. Each third-party vendor must adhere to strict quality and certification requirements before becoming qualified, thus ensuring superior support. Lattice pDS+ Fitters and their third-party tool partners provide seamless design environments to maximize designer productivity.

Lattice also offers innovative tools like the ISP Synario System which harnesses the power of Data I/O's Synario Entry, ABEL-HDL Compiler, and Functional Simulator and packages it in a complete logic design, fitter, and hardware solution.

ispCODE

Lattice provides a library of programming routines written in ANSI-standard C language (called ispCODE™) which can be easily incorporated into a system or tester software to support programming of ISP devices. These routines include such common operations as Program, Read, Verify, Erase, and Secure. After completion of the logic design and creation of a JEDEC file, in-system programming can be accomplished on customer-specific hardware: PCs, testers, or embedded systems. The ispCODE software package supplies specific routines, with extensively commented source code, for incorporation into application programs. These routines provide flexible, easy-to-use program modules which support the programming of a single device or multiple devices on a board. Example programs are included to demonstrate the use of each routine.

ISP Daisy Chain Download Software

Lattice also provides ISP Daisy Chain Download software and Windows-based executable utilities that support serial or parallel (turbo) programming of all Lattice ISP devices in a serial daisy chain configuration in the PC environment.

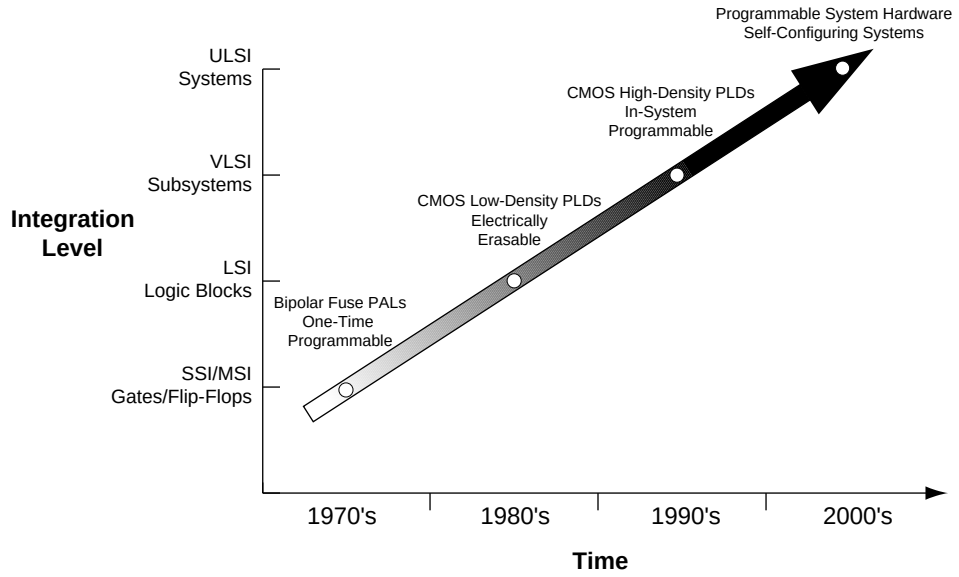
isp Engineering Kit and Download Cable

Lattice's isp Engineering Kits for ispLSI devices function as device programmers in conjunction with a PC. They can also be used for direct download to an ispLSI device on your board. The isp Engineering Kit interfaces with a PC and consists of a programming module, download cables and socket adapters to program any of the Lattice ispLSI devices. The ispDOWNLOAD™ cable supports programming of any ISP device directly on the PCB.

Future of ISP

In-system programmability is the logical evolution of programmable device technology. PLDs have evolved from the fuse-based, one-time programmable devices

Programmable Technology Evolution



invented in the 70s, to the electrically erasable components of the 80s, and now the in-system programmable devices of the 90s. The time and expense benefits of employing ISP make its widespread use inevitable.

But what about the future applications that ISP opens up?

To begin with, system designers will need to adopt a new mindset to exploit hardware that can evolve after the product is shipped. No longer will a hardware design be "frozen" as it is today; features will change based on updates transmitted from a central site or the system microprocessor can reconfigure peripheral functions in response to application needs. The use of ISP will drive hardware designs to more generic configurations that are given their "personality" through in-system programmable logic and interconnect components. Entire boards will be able to reconfigure microprocessor, memory and peripheral functions for any application. Just as a PC can run CAE tools, financial spreadsheets, games and multimedia software depending on the need, ISP hardware will be able to solve a broad range of problems just by downloading a new personality.

Ultimately, ISP will make the term "hardware" an anachronism, and usher in the era of truly programmable systems.

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