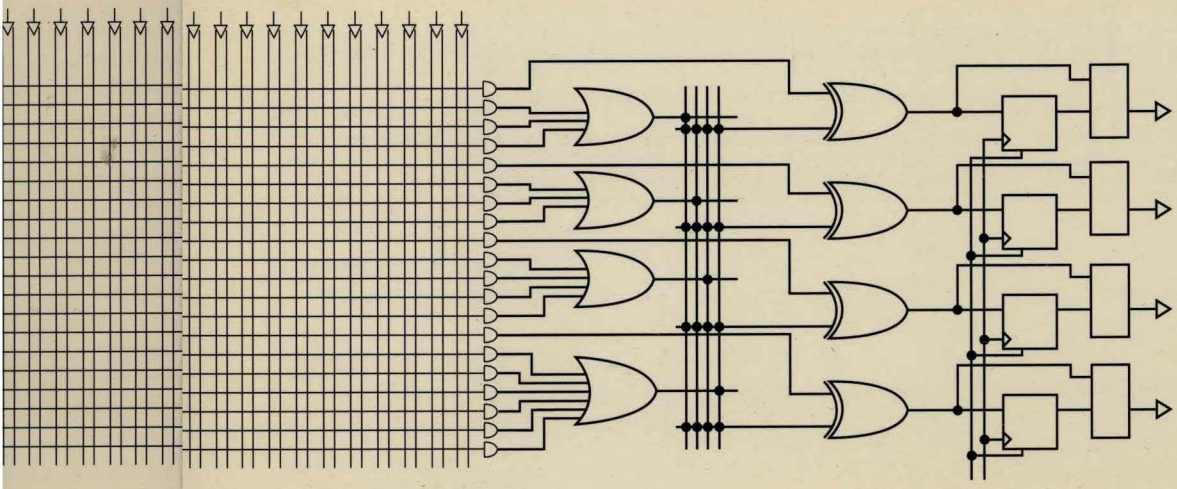
pLSI™ and ispLSI™
Data Book and
Handbook

Lattice

1992

1992
pLSI and ispLSI Data Book and Handbook



Lattice

pLSI and ispLSI Product Index

Commercial Grade Devices

pLSI Family – programmable Large Scale Integration

DEVICE	t_{pd}	f_{max}	DESCRIPTION	PAGE
pLSI 1016	15, 20	80, 50	44-pin programmable Large Scale Integration Device	2-31
pLSI 1024	15, 20	80, 50	68-pin programmable Large Scale Integration Device	2-53
pLSI 1032	15, 20	80, 50	84-pin programmable Large Scale Integration Device	2-1
pLSI 1048	20	70	120-pin programmable Large Scale Integration Device	2-75

ispLSI Family – in-system programmable Large Scale Integration

DEVICE	t_{pd}	f_{max}	DESCRIPTION	PAGE
ispLSI 1016	15, 20	80, 50	44-pin in-system programmable Large Scale Integration Device	2-129
ispLSI 1024	15, 20	80, 50	68-pin in-system programmable Large Scale Integration Device	2-155
ispLSI 1032	15, 20	80, 50	84-pin in-system programmable Large Scale Integration Device	2-95
ispLSI 1048	20	70	120-pin in-system programmable Large Scale Integration Device	2-181

Errata Sheet
1992 pLSI and ispLSI
Data Book and Handbook

Correction to Pages 2-4, 2-34, 2-56, 2-78

The NC pin Description should add:
"This pin should never be tied to GND".

Correction to pages 2-100, 2-134, 2-160, 2-186

The DC Electrical Characteristics table should have the following line added:

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNITS
I_{IL}-isp	$\overline{\text{ispEN}}$ input low current	$0V \leq V_{IN} \leq V_{IL}$ (MAX.)	—	—	-150	µA

Correction to page 3-35

t_{su} should read 6ns for -10 and 8ns for -15.
f_{max} (Maximum Clock Frequency with External Feedback, 1/(tsu + tco)) should read 76.9 MHz for -10 and 62.5 MHz for -15.

Correction to page 4-28

Instructions 01100 (GLBRLD) and 01101 (IOPRLD) are not supported at this time.

Correction to page 4-33

The Source Code listing is not complete, the correct version can be obtained on the pLSI Support BBS by registered users.

Correction to page 8-37

The last sentence in the Conclusion should read:
"The following section lists the Lattice Design File (LDF) with Boolean Equations and pinout for the ispLSI 1032".

Effective Date: February 1992

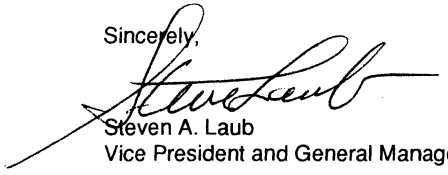
Thank you for your interest in our high density pLSI™ and ispLSI™ product families.

As the inventor and world wide market leader of the GAL® devices and E²CMOS® PLDs, we at Lattice are dedicated to supporting you with the fastest, highest quality and most innovative solutions to your programmable logic needs. We are reaffirming our commitment by offering our new high density product families of pLSI and ispLSI devices.

In this 1992 pLSI and ispLSI Data Book and Handbook, we have substantially broadened our product line by adding the world's highest performance and most flexible high density programmable logic devices.

We look forward to satisfying all of your programmable logic requirements.

Sincerely,



Steven A. Laub
Vice President and General Manager

pLSI and ispLSI Data Book and Handbook

1992

pLSI, ispLSI, Generic Array Logic, Latch-Lock, and RFT are trademarks of Lattice Semiconductor Corporation.
ispGAL, GAL, E^2CMOS and UltraMOS are registered trademarks of Lattice Semiconductor Corporation.

ABEL is a trademark of Data I/O Corporation.
Windows is a trademark and Microsoft is a registered trademark of Microsoft Corporation.
PAL is a registered trademark of Advanced Micro Devices, Inc.
Viewlogic and Viewism are registered trademarks of Viewlogic Systems, Inc.
Macintosh is a registered trademark of Apple Computers, Inc.

Products discussed in this literature are covered by U.S. Patents No. 4,761,768, 4,766,569, 4,833,646, 4,852,044, 4,855,954, 4,879,688, 4,887,239 and 4,896,296 issued to Lattice Semiconductor Corporation, and by U.S. and foreign patents pending.

LATTICE SEMICONDUCTOR CORP.
5555 Northeast Moore Court
Hillsboro, Oregon 97124 U.S.A.
Tel.: (503) 681-0118
FAX: (503) 681-3037
TELEX 277338 LSC UR
HOTLINE (800) LATTICE

Section 1: Introduction to pLSI™ and ispLSI™

Introduction to pLSI and ispLSI 1-1
Family Overview 1-1
pLSI and ispLSI Architecture 1-3
in-system programmability 1-3
pLSI/ispLSI Development System (pDS™) 1-3
Programming Support 1-4
Key pLSI and ispLSI Features 1-5

Section 2: pLSI and ispLSI Data Sheets

pLSI and ispLSI Product Index 2-ii
pLSI 1032 2-1
pLSI 1016 2-31
pLSI 1024 2-53
pLSI 1048 2-75
ispLSI 1032 2-95
ispLSI 1016 2-129
ispLSI 1024 2-155
ispLSI 1048 2-181

Section 3: GAL® Data Specifications

GAL Product Index 3-ii
Introduction to Generic Array Logic™ 3-1
GAL16V8A/B 3-3
GAL20V8A/B 3-9
GAL18V10 3-15
GAL22V10/B 3-19
GAL26CV12 3-25
GAL20RA10 3-29
GAL20XV10B 3-33
GAL6001 3-37
ispGAL16Z8 3-41
Military GAL Device Summary 3-45

Section 4: pLSI and ispLSI Architecture

Introduction 4-1
General Block Diagram 4-1
Generic Logic Block 4-3
I/O Cell 4-6
Output Routing Pool 4-8
Global Routing Pool 4-10
Megablock 4-10
Clock Distribution Network 4-12
Timing Model 4-14
ispLSI Programming Information 4-23

Section 5: pLSI and ispLSI Advantages

Introduction 5-1
Advantages of the Architecture 5-1
Advantages of E^2CMOS Technology 5-2

Section 5: pLSI and ispLSI Advantages (Continued)
E²CMOS Advantages Over Other Technologies 5-3
Advantages of in-system programming 5-4
Advantages of the pLSI and ispLSI Design Tools 5-6
Section 6: pLSI and ispLSI Software Development Tools
System Design Process 6-1
pLSI and ispLSI Design Flow 6-3
pLSI and ispLSI Development System 6-5
The User Interface 6-5
Entering a Design 6-8
Verifying a Design 6-10
Lattice Place and Route 6-11
Generating the Programming File 6-13
Download to a Device 6-14
Report Files 6-14
Clock Distribution Network 6-20
Macro Usage 6-22
Logic Simulation of a Design 6-27
Section 7: pLSI and ispLSI Design Optimization
GLB Partitioning 7-1
Placement and Routing 7-9
Device Utilization 7-17
Speed 7-21
Section 8: pLSI and ispLSI Application Notes
Selecting the Right High Density Device 8-1
Beginner's Guide 8-7
A Digital Clock Design Example 8-21
ispLSI Configurable Memory Controller 8-33
pLSI and ispLSI: A Multiple Function Solution 8-47
Video Graphics Controller 8-67
Compiling Multiple PLDs into a pLSI Device 8-91
Section 9: pLSI and ispLSI Article Reprints
Fast, High-Density PLDs Offer On-Board Reprogramming 9-1
Lattice Fields FPGA 9-3
Avoid The Pitfalls Of High-Speed Logic Design 9-5
Multiple Factors Define True Cost of PLDs 9-11
Section 10: pLSI and ispLSI Quality, Reliability & Programmer Support
Quality Assurance Program 10-1
Qualification Program 10-3
E²CMOS Testability Improves Quality 10-4
Package Thermal Resistance 10-5
Package Diagrams 10-6
pLSI and ispLSI 1032 Programmer Support 10-8
Section 11: Sales Information
Sales Offices 11-1

Section 1: Introduction to pLSI™ and ispLSI™

Introduction to pLSI and ispLSI 1-1

Family Overview 1-2

The pLSI and ispLSI Architecture 1-2

in-system programmability 1-5

pLSI/ispLSI Development System (pDS™) 1-7

Programming Support 1-8

Key pLSI and ispLSI Features 1-8

Section 2: pLSI and ispLSI Data Sheets

Section 3: GAL® Data Specifications

Section 4: pLSI and ispLSI Architecture

Section 5: pLSI and ispLSI Advantages

Section 6: pLSI and ispLSI Software Development Tools

Section 7: pLSI and ispLSI Design Optimization

Section 8: pLSI and ispLSI Application Notes

Section 9: pLSI and ispLSI Article Reprints

Section 10: pLSI and ispLSI Quality, Reliability & Programmer Support

Section 11: Sales Information

Introduction to pLSI™ and ispLSI™

Introduction to pLSI and ispLSI

Lattice Semiconductor's pLSI (programmable Large Scale Integration) and ispLSI (in-system programmable Large Scale integration) are two families of high density and high performance $E^2CMOS^{®}$ programmable logic devices (see figure 1-1). They provide design engineers with a superior system solution for integrating high speed logic features on a single chip.

The Lattice pLSI and ispLSI families are the first programmable logic devices to combine the performance and ease of use of PLDs with the density and flexibility of FPGAs.

The ispLSI family also pioneers non-volatile, in-system programmability, a technology that allows real-time programming, less expensive manufacturing and end-user reconfiguration.

Lattice's E^2CMOS technology features reprogrammability, the ability to program the device again and again to easily incorporate any design modifications. This same capability allows full parametric testability during manufacturing, which guarantees 100 percent programming and functional yield.

All the necessary development tools are available from Lattice and leading third-party companies. Utilizing a Windows-based graphical user interface, it is possible to complete a circuit design in hours, as opposed to weeks or months.

pLSI and ispLSI Product Families

- ❑ 80 MHz System Performance
- ❑ 15 ns Pin-to-Pin Delay
- ❑ Deterministic Performance
- ❑ High Density (2,000-8,000 PLD Gates)
- ❑ Flexible Architecture
- ❑ Easy to Use
- ❑ in-system programmable (ispLSI)
- ❑ Low Power Consumption

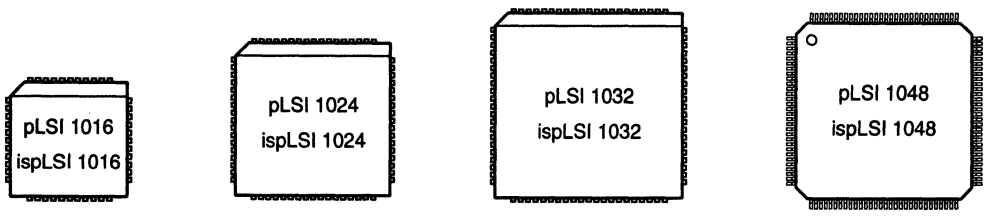
pLSI and ispLSI Technology

- ❑ E^2CMOS — the PLD Technology of Choice
- ❑ Proven UltraMOS-IV Technology (0.8 micron) Feature Size
- ❑ Electrically Erasable/Programmable/Reprogrammable
- ❑ 100% Tested During Manufacture
- ❑ 100% Programming Yield

pLSI and ispLSI Development Tools

- ❑ Easy-to-Use Graphical Interface (Windows 3.0)
- ❑ Boolean Equations and Macro Input
- ❑ Industry-Standard Third-Party Design Environment and Platforms

Figure 1-1. pLSI and ispLSI Device Families



44-Pin PLCC | 68-Pin PLCC | 84-Pin PLCC | 120-Pin PQFP

Family Overview

The pLSI and ispLSI families of high-density devices address high-performance system logic needs, ranging from registers, to counters, to multiplexers, to complex state machines.

With PLD gate densities ranging from 2,000 to 8,000, the pLSI and ispLSI families provide a range of programmable logic solutions to meet design requirements for today's and tomorrow's needs.

Each device contains multiple logic blocks (GLBs), architected to maximize system flexibility and performance. A balanced ratio of registers and I/O cells provides the optimum combination of internal logic and external connections. A global interconnect scheme ties everything together, enabling utilization of more than 80% of available logic. Table 1-1 describes the family attributes.

The pLSI and ispLSI Architecture

The pLSI and ispLSI architecture was constructed with actual system design requirements in mind. This architecture provides the designer with the following advantages. Figure 1-2 shows the pLSI 1032 architecture.

- High Speed
- Predictable Performance
- Integration of Multiple Logic Functions
- Asynchronous Designs
- Flexible Logic Paths
- Advanced Global Clock Network

The Global Routing Pool (GRP)

Central to the pLSI and ispLSI architecture is the Global Routing Pool, which connects all of the internal logic and makes it available to the designer. The GRP provides complete interconnectivity with fixed and predictable delays. This unique connection scheme consistently provides high performance and allows effortless implementation of complex designs.

The Output Routing Pool (ORP)

Pin assignment flexibility is maximized via the Output Routing Pool (ORP), which provides the connections between the GLB outputs and the output pins.

Figure 1-2. pLSI 1032 Architecture

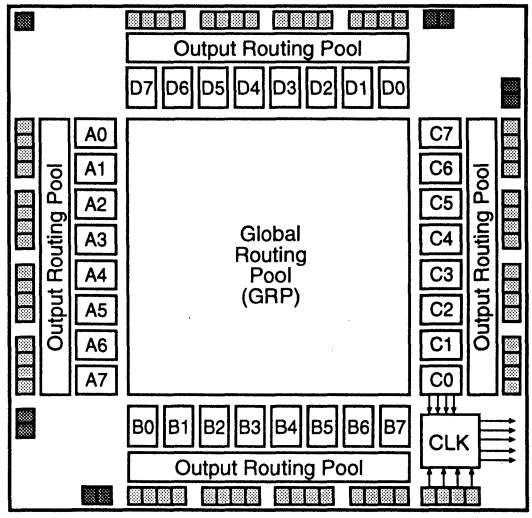


Table 1-1. pLSI and ispLSI Family Attributes

Family Member	1016	1024	1032	1048
Density	2,000	4,000	6,000	8,000
Speed: f_{max} (MHz)	80	80	80	70
Speed: t_{pd} (ns)	15	15	15	20
GLBs	16	24	32	48
Registers	96	144	192	288
I/O	36	54	72	106
Pin/Package	44-pin PLCC	68-pin PLCC	84-pin PLCC	120-pin PQFP

Generic Logic Block (GLB)

The basic logic element in the pLSI and ispLSI architecture is the Generic Logic Block. This powerful logic element provides an input-to-output ratio greater than 4:1. With 18 inputs driving an array of 20 product terms (PTs) — which in turn feed four outputs — the GLB efficiently handles both wide and narrow gating functions. Figure 1-3 describes the GLB functionality.

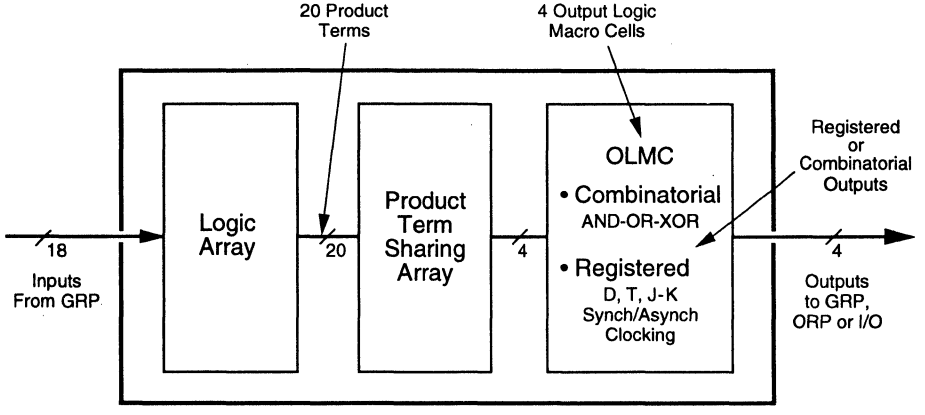
One element of architectural flexibility is the Product Term Sharing Array. The PTSA allows the 20 Product Terms (PTs) from the AND array to be shared with any and all of the four GLB outputs as needed to implement logic designs. This ability to share PTs between all of the GLB outputs provides a highly efficient implementation of complex state machines by eliminating duplicate product term groups.

The architecture flexibility of the GLB, combined with its optimum input-to-output ratio, allows the GLB to implement virtually all 4-bit MSI functions.

Each of the four outputs from the PTSA feeds into a flexible Output Logic Macrocell (OLMC), consisting of a D-type flip-flop with an Exclusive-OR gate on the input. The OLMC allows each GLB output to be configured either combinatorial or registered. Combinatorial mode is available as AND-OR or Exclusive-OR; registered mode is available as D, T or J-K.

The GLB can be clocked synchronously or asynchronously. Global clocks from external pins or internally generated, provide all GLBs and I/O Cells with synchronous clock signals with selectable polarity. This provides multiple synchronous clock phases to all GLBs and I/Os.

Figure 1-3. Simplified Generic Logic Block Functionality



The GLB has several configuration options for each Output Logic Macrocell (OLMC). These can be mixed with each GLB. The configurations are described as standard, high-speed bypass, XOR and multi-mode configuration.

Standard Configuration

- GLB Outputs Comprise of 4,4,5 or 7 Product Terms
- The PTSA Can Combine up to 20 PTs per GLB Output to Meet the Needs of Both Wide and Narrow Logic Functions.

High-Speed Bypass Configuration

- For Speed-Critical Timing Paths
- Enables Design of Fast Address Decoders
- Bypasses the PTSA and the Internal Exclusive-OR Gate of the OLMC
- Provides Four Product Terms Per Output

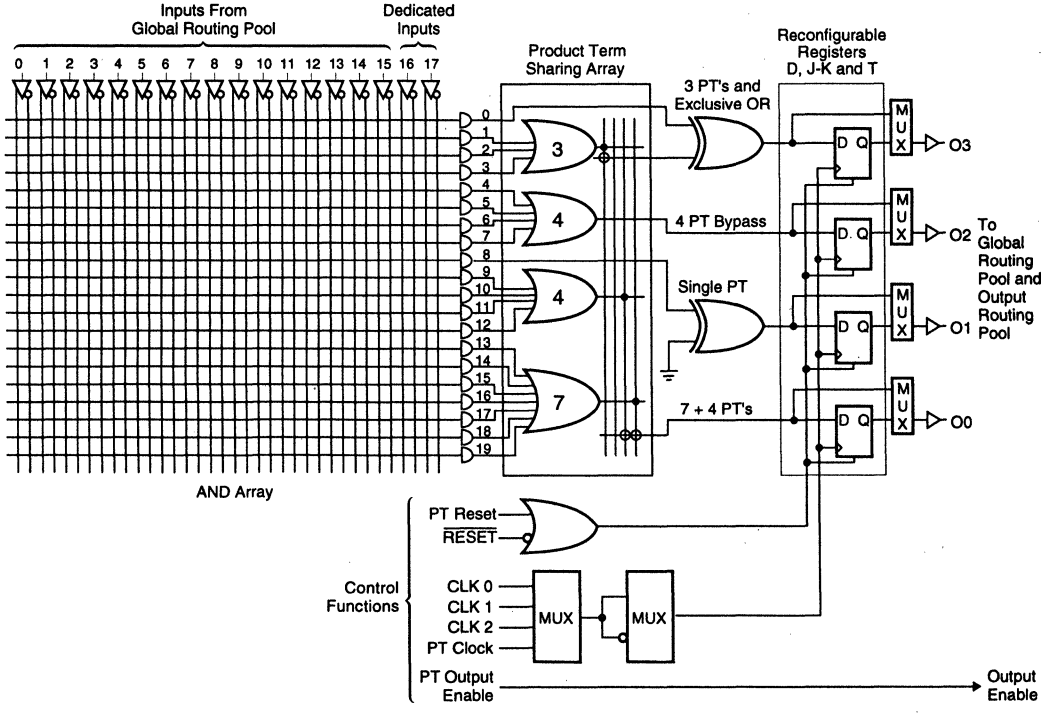
XOR Configuration

- Utilizes Powerful Exclusive-OR Architecture
- Powerful for Counters, Comparators and ALU Functions

Multi-Mode Configuration

- Individual Outputs are Independently Configurable
- PTSA Allows Flexibility on the Number and Selection of Product Terms Per Output

Figure 1-4. GLB: Multi-Mode Configuration



in-system programmability

The in-system programmable Large Scale Integration (ispLSI) family is the industry's only high-density programmable logic family offering non-volatile in-system reconfigurability.

The ispLSI family is 100 percent functionally and parametrically compatible with the pLSI family, with the added ability of 5-volt in-system programmability and reprogrammability.

Complex logic functions can be implemented in multiple ispLSI devices, with complete on-board configurability. In-system programming of multiple ispLSI chip solutions is easily achieved through a proprietary in-system erase/program/verify technique.

In-system programmability can revolutionize the way boards are designed, manufactured and serviced (see figure 1-5).

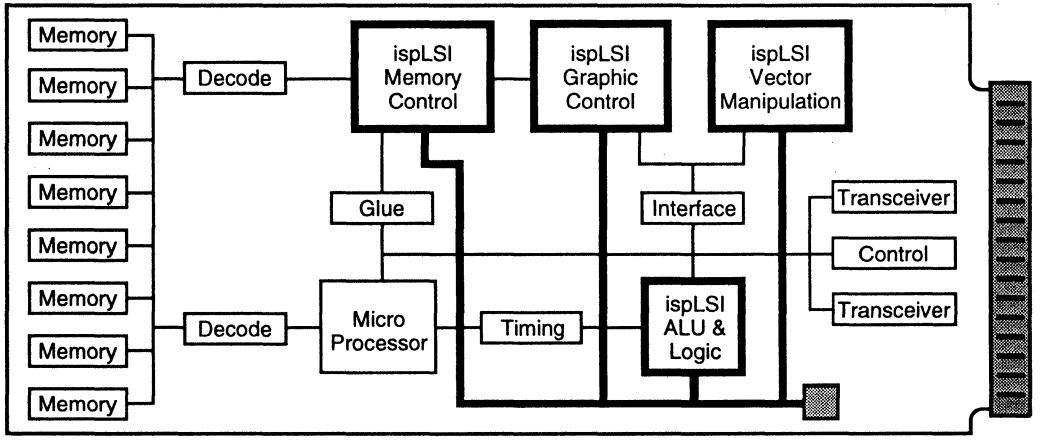
Prototype board designs - in-system programming allows the programming and modification of logic designs "in-system" without removing the device(s) from the board. This accelerates the system and board-level debug process and enables definition of board layout earlier in the design process.

Reconfigurable systems - The options for accommodating changes are greatly increased when you have the ability to change the functionality of devices already soldered on a board. Multiple hardware configurations can be implemented with the same circuit board design. Multiple protocols or multiple system interfaces can be defined on a generic board as the last step in the manufacturing flow.

Diagnostic Capability - Using the ispLSI device, the diagnostic capability of the system can be enhanced. A test pattern can be programmed into the ispLSI device at board-test, enabling the logic to control and observe specific nodes of the entire board. After the diagnostic testing is complete, the functional pattern can be programmed into the device for normal system operation.

Easier field updates - With software reconfigurable systems, field updates are as easy as loading a new device configuration from a floppy, or downloading it through a modem.

Figure 1-5. in-system programmable "Generic" Board



Multiple ispLSI devices can be reconfigured through multiplexed signals interfaced via an edge connector, 5-post connector, microcontroller, or microprocessor.

A powerful benefit of the ispLSI family is its potential to streamline the manufacturing process by eliminating the separate programming and labeling steps usually associated with PLDs. Quality is enhanced when product handling steps are reduced, in this case, those associated with programming, labeling and re-inventorying multiple device types. Eliminating socketing further improves quality and reduces board cost. Figure 1-6 shows the enhanced manufacturing with the ispLSI device.

All necessary programming is achieved via five TTL-level logic interface signals (see figure 1-7). These five signals control the on-chip programming circuitry, which is securely protected against inadvertent reprogramming via on-chip state machines. The ispLSI family can also be programmed using popular third-party logic programmers.

Figure 1-6. Manufacturing Flow Comparison

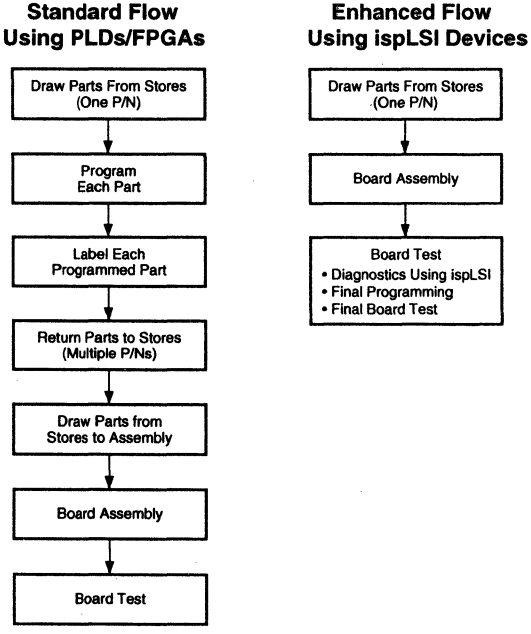
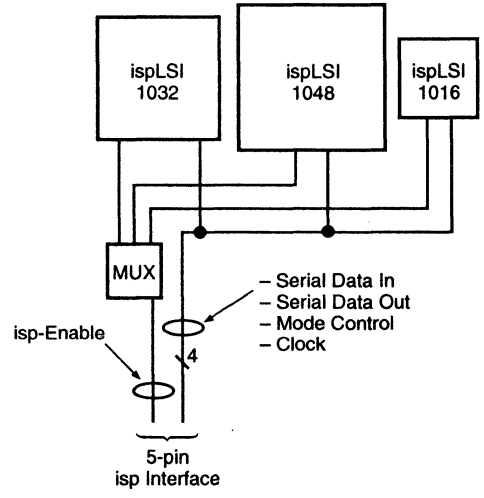


Figure 1-7. in-system programming Interface (Multi-Chip Solution)



pLSI/ispLSI Development System (pDS™)

Both the pLSI and ispLSI families are supported by Lattice's pLSI/ispLSI Development System. It runs on IBM-compatible (386/486) PCs with Microsoft® Windows version 3.0.

The easy-to-use graphical-user interface (see figure 1-8) with familiar mouse and pull-down menus, combined with Boolean Equations entry (using ABEL®-like syntax) allows immediate design productivity with pLSI and ispLSI devices.

The Windows graphical user interface makes design-entry easy, using pull-down menus, intuitive point-and-click commands and self-explanatory instructions. Without any up-front training, designs can be completed in hours instead of weeks or months.

The pDS Software supports over 200 Macros to help speed the design process. These Macros cover most TTL functions, from gate primitives to 16-bit counters. Lattice pDS Software also supports user-definable Macros, which can be modifications of existing Macros or custom creations.

The Lattice Place and Route allows assignment of pins and critical speed paths, and ensures optimized 100% routability at 80% utilization.

Quick compilation speeds the design, debug and rework process dramatically. pDS software also supports incremental design techniques.

Timing and functional simulation is also available from Lattice, using Viewlogic's Viewsim® simulation software. The design flow with the pDS Software is described in figure 1-9.

Figure 1-8. pDS Software Graphical User Interface

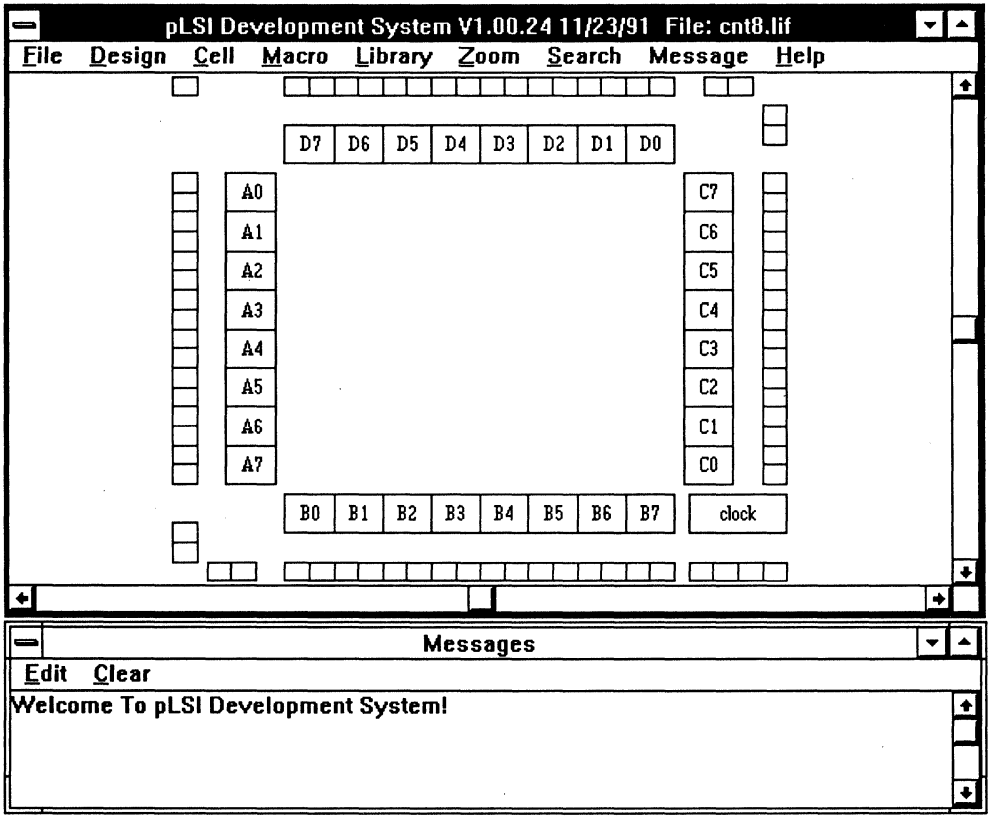


Figure 1-8. pDS Design Flow

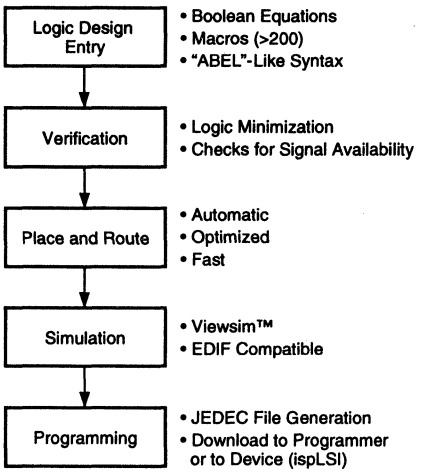


Table 1-2. Programming Support

Programmer Vendor	Model
Advin Systems	Pilot GL/U40
	Pilot U84
BP Micro	PLD1128
Data I/O	2900
	3900
	Unisite 40/48
Logical Devices	Allpro 32/40
	Allpro 88
SMS Microsystems	Sprint Expert
Stag	ZL30A
System General	Turpro 1

Programming Support

The pLSI and ispLSI families are supported by popular third-party logic programmers including Data I/O, Logical Devices, Stag, System General, SMS Microcomputer and Advin. Table 1-2 describes each vendor's specific programmer model that support the pLSI and ispLSI devices. No proprietary, expensive, high pin-count programmers are required. Additionally, the ispLSI family can be programmed on the board (in-system), which eliminates the need for a stand-alone programmer.

Key pLSI and ispLSI Features

- ❑ Predictable High-Speed System Performance
  - 80 MHz System Speed
  - 15 ns t_{pd}
- ❑ High Density (2,000 to 8,000 PLD Gates)
- ❑ Flexible, Powerful Architecture
  - Glue Logic to Counters to State Machines
- ❑ in-system programmability and Reprogrammability
- ❑ Easy-to-Use Development Software
  - Familiar (ABEL-Like)
  - Fast (Minutes)
  - Automatic (No Manual Intervention Required)
  - 100% Routing With Greater than 80% Utilization

Section 1: Introduction to pLSI™ and ispLSI™

Section 2: pLSI and ispLSI Data Sheets

pLSI and ispLSI Product Index 2-ii

pLSI 1032 2-1

pLSI 1016 2-31

pLSI 1024 2-53

pLSI 1048 2-75

ispLSI 1032 2-95

ispLSI 1016 2-129

ispLSI 1024 2-155

ispLSI 1048 2-181

Section 3: GAL® Data Specifications

Section 4: pLSI and ispLSI Architecture

Section 5: pLSI and ispLSI Advantages

Section 6: pLSI and ispLSI Software Development Tools

Section 7: pLSI and ispLSI Design Optimization

Section 8: pLSI and ispLSI Application Notes

Section 9: pLSI and ispLSI Article Reprints

Section 10: pLSI and ispLSI Quality, Reliability & Programmer Support

Section 11: Sales Information

Commercial Grade Devices

pLSI Family – programmable Large Scale Integration

DEVICE	t_{pd}	f_{max}	DESCRIPTION	PAGE
pLSI 1016	15, 20	80, 50	44-pin programmable Large Scale Integration Device	2-31
pLSI 1024	15, 20	80, 50	68-pin programmable Large Scale Integration Device	2-53
pLSI 1032	15, 20	80, 50	84-pin programmable Large Scale Integration Device	2-1
pLSI 1048	20	70	120-pin programmable Large Scale Integration Device	2-75

ispLSI Family – in-system programmable Large Scale Integration

DEVICE	t_{pd}	f_{max}	DESCRIPTION	PAGE
ispLSI 1016	15, 20	80, 50	44-pin in-system programmable Large Scale Integration Device	2-129
ispLSI 1024	15, 20	80, 50	68-pin in-system programmable Large Scale Integration Device	2-155
ispLSI 1032	15, 20	80, 50	84-pin in-system programmable Large Scale Integration Device	2-95
ispLSI 1048	20	70	120-pin in-system programmable Large Scale Integration Device	2-181

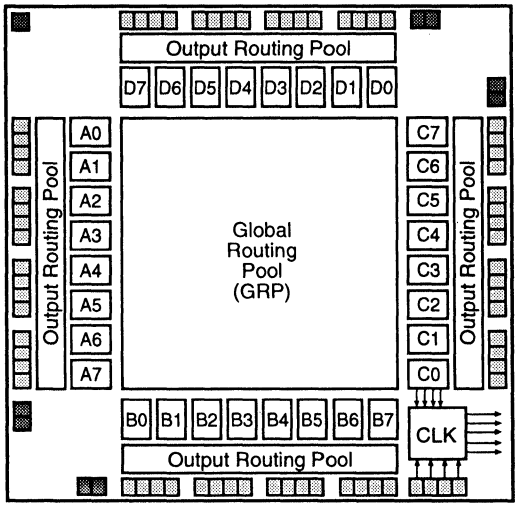
pLSI™ 1032

programmable Large Scale Integration

Features

- **PROGRAMMABLE HIGH DENSITY LOGIC**
  - — Member of Lattice's pLSI Family
  - — High Speed Global Interconnects
  - — 64 I/O Pins, Eight Dedicated Inputs
  - — 192 Registers
  - — Wide Input Gating for Fast Counters, State Machines, Address Decoders, etc.
  - — Small Logic Block Size for Random Logic
  - — Security Cell Prevents Unauthorized Copying
- **HIGH PERFORMANCE E²CMOS® TECHNOLOGY**
  - — f_{max} = 80 MHz Maximum Operating Frequency
  - — t_{pd} = 15 ns Propagation Delay
  - — Low Power Consumption (I_{cc} 135mA Typ.)
  - — TTL Compatible Inputs and Outputs
  - — Electrically Erasable and Re-Programmable
  - — 100% Tested
- **COMBINES EASE OF USE AND THE FAST SYSTEM SPEED OF PLDs WITH THE DENSITY AND FLEXIBILITY OF FIELD PROGRAMMABLE GATE ARRAYS**
  - — Complete Programmable Device can Combine Glue Logic and Structured Designs
  - — 100% Routable at 80% Utilization
  - — Four Dedicated Clock Input Pins
  - — Synchronous and Asynchronous Clocks
  - — Flexible Pin Placement
  - — Optimized Global Routing Pool Allows Global Interconnectivity
- **pLSI/ispLSI™ DEVELOPMENT SYSTEM (pDS™)**
  - — Boolean Logic Compiler
  - — Automatic Place and Route
  - — Manual Partitioning
  - — PC Platform
  - — Easy to Use Windows Interface
- **ADVANCED pLSI/ispLSI DEVELOPMENT SYSTEM**
  - — Industry Standard, Third Party Design Environments
  - — Schematic Capture
  - — Fully Automatic Partitioning
  - — Automatic Place and Route
  - — Comprehensive Logic and Timing Simulation
  - — PC and Workstation Platforms

Functional Block Diagram



Description

The Lattice pLSI 1032 is a High Density Programmable Logic Device which contains 192 Registers, 64 Universal I/O pins, eight Dedicated Input Pins, four Dedicated Clock Input Pins and a Global Routing Pool (GRP). The GRP provides complete interconnectivity between all of these elements.

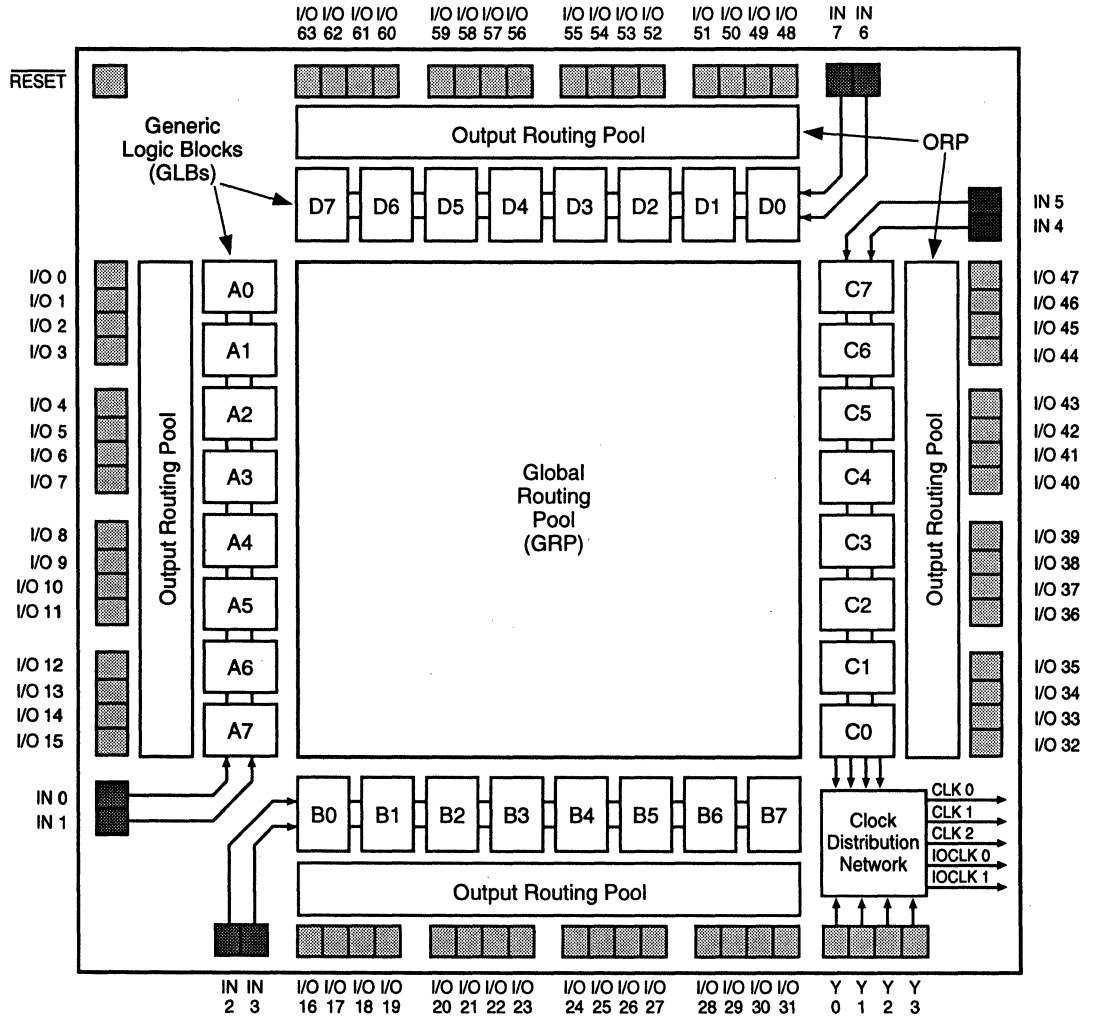
The basic unit of logic on the pLSI 1032 device is the Generic Logic Block (GLB). The GLBs are labeled A0, A1 .. D7, (see figure 1). There are a total of 32 GLBs in the pLSI 1032 device. Each GLB has 18 inputs, a programmable AND/OR/Exclusive OR array, and four outputs which can be configured to be either combinatorial or registered. Inputs to the GLB come from the GRP. All of the GLB outputs are brought back into the GRP so that they can be connected to the inputs of any other GLB on the device.

LATTICE SEMICONDUCTOR CORP., 5555 Northeast Moore Ct., Hillsboro, Oregon 97124, U.S.A.
Tel. 1-800-LATTICE (528-8423); FAX (503) 681-3037

January 1992. Rev. A

Functional Block Diagram

Figure 1. pLSI 1032



Description (continued)

The device also has 64 I/O Cells, each of which is directly connected to an I/O pin. Each I/O cell can be individually programmed to be a combinatorial input, registered input, latched input, output or bi-directional I/O pin with 3-state. The signal levels are TTL compatible voltages and the output drivers can source 4 mA or sink 8 mA.

The 64 I/O Cells are grouped into four sets of 16 each as shown in figure 1. Each of these I/O groups is associated with a logic Megablock through the use of the Output Routing Pool (ORP) and shares a common Output Enable (OE) signal.

Eight GLBs, 16 I/O Cells and one ORP are connected together to make a logic Megablock. The Megablock is defined by the resources that it shares. The outputs of the eight GLBs are connected to a set of 16 universal I/O cells by the ORP. The pLSI 1032 Device contains four of these Megablocks.

The GRP has as its inputs the outputs from all of the GLBs and all of the inputs from the bi-directional I/O cells. All of these signals are made available to the inputs of the GLBs. Delays through the GRP have been equalized to minimize timing skew and logic glitching.

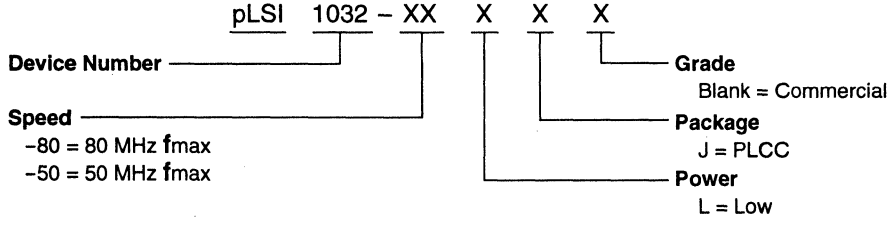
Clocks in the pLSI 1032 device are selected using the Clock Distribution Network. Four dedicated clock pins (Y0 to Y3) are brought into the distribution network, and five outputs (CLK 0 to CLK 2 and IOCLK 0, IOCLK 1) are provided to route clocks to the GLBs and I/O cells. The Clock Distribution Network can also be driven from a special GLB (C0 on the pLSI 1032 device). The logic of this GLB allows the user to create an internal clock from a combination of internal signals within the device.

The pLSI 1032 device is part of Lattice's programmable Large Scale Integration (pLSI) family. This family contains a range of devices from the pLSI 1016, with 96 registers, to the pLSI 1048 with 288 registers. The pLSI Family Product Selector Guide below lists key attributes of the devices along with the number of resources available.

pLSI Family Product Selector Guide

DEVICE	pLSI 1016	pLSI 1024	pLSI 1032	pLSI 1048
GLBs	16	24	32	48
Registers	96	144	192	288
I/O Pins	32	48	64	96
Dedicated Inputs	4	6	8	10
Pin Count	44	68	84	120

Ordering Information



Pin Description

Name	PLCC Pin Numbers	Description
I/O 0 - I/O 3 I/O 4 - I/O 7 I/O 8 - I/O 11 I/O 12 - I/O 15 I/O 16 - I/O 19 I/O 20 - I/O 23 I/O 24 - I/O 27 I/O 28 - I/O 31 I/O 32 - I/O 35 I/O 36 - I/O 39 I/O 40 - I/O 43 I/O 44 - I/O 47 I/O 48 - I/O 51 I/O 52 - I/O 55 I/O 56 - I/O 59 I/O 60 - I/O 63	26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18	Input/Output Pins - These are the general purpose I/O pins used by the logic array.
IN 0 - IN 3 IN 4 - IN 7	25, 42, 44, 61 67, 84, 2, 19	Dedicated input pins to the device.
$\overline{\text{RESET}}$	24	Active Low (0) Reset pin which resets all of the GLB and I/O registers in the device.
Y0	20	Dedicated Clock input. This clock input is connected to one of the clock inputs of all of the GLBs on the device.
Y1	66	Dedicated Clock input. This clock input is brought into the clock distribution network, and can optionally be routed to any GLB on the device.
Y2	63	Dedicated Clock input. This clock input is brought into the clock distribution network, and can optionally be routed to any GLB and/or any I/O Cell on the device.
Y3	62	Dedicated Clock input. This clock input is brought into the clock distribution network, and can optionally be routed to any I/O Cell on the device.
NC	23	This is a factory test pin and it should be left floating or tied to V_{CC}
GND VCC	1, 22, 43, 64 21, 65	Ground (GND) V_{CC}

Absolute Maximum Ratings [1]

Supply Voltage V_{CC} . . . . . -0.5 to +7.0V

Input Voltage Applied. . . . . -2.5 to V_{CC} +1.0V

Off-State Output Voltage Applied . . . . . -2.5 to V_{CC} +1.0V

Storage Temperature . . . . . -65 to 125°C

Ambient Temp. with Power Applied . . . . . -55 to 125°C

1. Stresses above those listed under the "Absolute Maximum Ratings" may cause permanent damage to the device. Functional operation of the device at these or at any other conditions above those indicated in the operational sections of this specification is not implied (while programming, follow the programming specifications).

DC Recommended Operating Condition

SYMBOL	PARAMETER	MIN.	MAX.	UNITS
T_A	Ambient Temperature	0	70	°C
V_{CC}	Supply Voltage	4.75	5.25	V
V_{IL}	Input Low Voltage	0	0.8	V
V_{IH}	Input High Voltage	2.0	V_{CC}	V

Capacitance (T_A=25°C,f=1.0 MHz)

SYMBOL	PARAMETER	MAXIMUM[1]	UNITS	TEST CONDITIONS
C_1	Input Capacitance	8	pf	V_{CC}=5.0V, V_{IN}=2.0V
C_2	I/O, Y Capacitance	10	pf	V_{CC}=5.0V, VI/O, Y=2.0V

1. Guaranteed but not 100% tested.

Data Retention Specifications

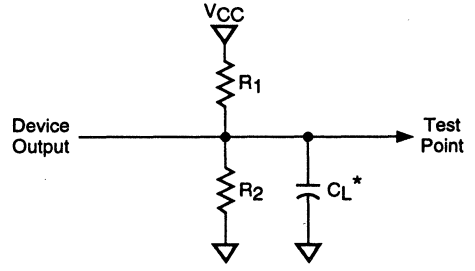
PARAMETER	MINIMUM	MAXIMUM	UNITS
Data Retention	20	–	YEARS
Erase/Reprogram Cycles	–	100	CYCLES

Switching Test Conditions

Input Pulse Levels	GND to 3.0V
Input Rise and Fall Times	3ns 10% to 90%
Input Timing Reference Levels	1.5V
Output Timing Reference Levels	1.5V
Output Load	See Figure 2

3-state levels are measured 0.5V from steady-state active level.

Figure 2. Test Load



*C_L Indicates Test Fixture and Probe Total Capacitance

Output Load Conditions (see figure 2)

Test Condition		R1	R2	CL
1		470Ω	390Ω	35pF
2	Active High	∞	390Ω	35pF
	Active Low	470Ω	390Ω	35pF
3	Active High to Z at V_{OH} - 0.5V	∞	390Ω	5pF
	Active Low to Z at V_{OL} + 0.5V	470Ω	390Ω	5pF

DC Electrical Characteristics

Over Recommended Operating Conditions

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNITS
V_{OL}	Output Low Voltage	I_{OL} =8 mA.	–	–	0.4	V
V_{OH}	Output High Voltage	I_{OH} =-4 mA.	2.4	–	–	V
I_{IL}	Input or I/O Low Leakage Current	$0V \le V_{IN} \le V_{IL}$ (MAX.)	–	–	-10	µA
I_{IH}	Input or I/O High Leakage Current	$V_{IH} \le V_{IN} \le V_{CC}$	–	–	10	µA
I_{OS}[1]	Output Short Circuit Current	V_{CC} = 5V, V_{OUT} = 0.5V	-60	–	-200	mA
I_{CC}[2]	Operating Power Supply Current	V_{IL} = 0.5V, V_{IH} = 3.0V F_{TOGGLE} = 20 MHz	–	135	195	mA

1. One output at a time for a maximum duration of one second. (Vout = 0.5V)
2. Measured at a frequency of 20 MHz using eight 16-bit counters.

External Switching Characteristics[1, 2, 3] pLSI 1032-80

Over Recommended Operating Conditions

PARAMETER	TEST[6] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
t_{pd1}	1	1	Data Propagation Delay, 4PT bypass, ORP bypass	–	12	15	ns
t_{pd2}	1	2	Data Propagation Delay	–	15	20	ns
t_{co1}[5]	1	3	External Clock to Output Delay, ORP bypass	–	8	11	ns
t_{co2}[5]	1	4	External Clock to Output Delay	–	9	14	ns
t_{co3}	1	5	Internal Synch. Clock to Output Delay	–	15	20	ns
t_{co4}	1	6	Asynchronous Clock to Output Delay	–	13	20	ns
t_{r1}	1	7	External Pin Reset to Output Delay	–	13	20	ns
t_{r2}	1	8	Asynchronous PT Reset to Output Delay	–	15	22	ns
t_{en}	2	9	Input to Output Enable	–	13	20	ns
t_{dis}	3	10	Input to Output Disable	–	13	20	ns

External AC Recommended Operating Conditions[1, 2, 3] pLSI 1032-80

Over Recommended Operating Conditions

PARAMETER	TEST[6] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
f_{max}[4]	1	11	Clock Frequency with Internal Feedback	–	100	80	MHz
f_{max} (External)	1	12	Clock Frequency with External Feedback	–	70	50	MHz
t_{su1}	–	13	Setup Time before External Synch. Clock, 4PT bypass	9	6	–	ns
t_{su2}	–	14	Setup Time before External Synch Clock	12	8	–	ns
t_{su3}	–	15	Setup Time before Internal Synch. Clock	9	3	–	ns
t_{su4}	–	16	Setup Time before Asynchronous Clock	9	4	–	ns
t_{h1}	–	17	Hold time after External Synchronous Clock, 4PT bypass	2	-1	–	ns
t_{h2}	–	18	Hold time after External Synchronous Clock	2	-1	–	ns
t_{h3}	–	19	Hold time after Internal Synchronous Clock	8	2	–	ns
t_{h4}	–	20	Hold time after Asynchronous Clock	8	1	–	ns
t_{rw1}	–	21	External Reset Pulse Duration	10	8	–	ns
t_{rw2}	–	22	Asynchronous Reset Pulse Duration	10	8	–	ns
t_{wh1}, t_{wl1}	–	23,24	External Synchronous Clock Pulse Duration, High, Low	6	5	–	ns
t_{wh2}, t_{wl2}	–	25,26	Asynchronous Clock Pulse Duration, High, Low	6	5	–	ns

1. External Parameters are tested and guaranteed.
2. See Timing Technical Note for further details.
3. Unless noted, all parameters use ORP, GRP fanout of four, PTSA, and are measured with 16 outputs switching.
4. Standard 16-bit counter implementation using GRP feedback.
5. Clock to output specifications include a maximum skew of 2 ns.
6. Refer to Switching Test Conditions section.

Switching Characteristics[1, 2, 3] pLSI 1032-80

Using I/O Cell

PARAMETER	TEST[4] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
t_{su5}	–	27	Setup Time before External Synchronous Clock	5	0	–	ns
t_{su6}	–	28	Setup Time before Internal Synchronous Clock	0	-3	–	ns
t_{h5}	–	29	Hold Time after External Synchronous Clock	8	4	–	ns
t_{h6}	–	30	Hold Time after Internal Synchronous Clock	15	11	–	ns
t_{wh3}, t_{wl3}	–	31,32	Clock Pulse Duration, High, Low	6	5	–	ns

1. External Parameters are tested and guaranteed.
2. See Timing Technical Note for further details.
3. Unless noted, all parameters use ORP, GRP fanout of four, PTSA, and are measured with 16 outputs switching.
4. Refer to Switching Test Conditions section.

External Switching Characteristics[1, 2, 3] pLSI 1032-50

Over Recommended Operating Conditions

PARAMETER	TEST[6] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
t_{pd1}	1	1	Data Propagation Delay, 4PT bypass, ORP bypass	–	16	20	ns
t_{pd2}	1	2	Data Propagation Delay, ORP	–	19	25	ns
t_{co1}[5]	1	3	External Clock to Output Delay, ORP bypass	–	12	16	ns
t_{co2}[5]	1	4	External Clock to Output Delay	–	15	20	ns
t_{co3}	1	5	Internal Synch. Clock to Output Delay	–	21	28	ns
t_{co4}	1	6	Asynchronous Clock to Output Delay	–	21	28	ns
t_{r1}	–	7	External Pin Reset to Output Delay	–	21	28	ns
t_{r2}	–	8	Asynchronous PT Reset to Output Delay	–	24	30	ns
t_{en}	2	9	Input to Output Enable	–	21	28	ns
t_{dis}	3	10	Input to Output Disable	–	21	28	ns

External AC Recommended Operating Conditions[1, 2, 3] pLSI 1032-50

Over Recommended Operating Conditions

PARAMETER	TEST[6] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
f_{max}[4]	1	11	Clock Frequency with Internal Feedback	–	70	50	MHz
f_{max} (External)	1	12	Clock Frequency with External Feedback	–	45	33	MHz
t_{su1}	–	13	Setup Time before External Synch. Clock, 4PT bypass	14	10	–	ns
t_{su2}	–	14	Setup Time before External Synch Clock	17	13	–	ns
t_{su3}	–	15	Setup Time before Internal Synch. Clock	13	9	–	ns
t_{su4}	–	16	Setup Time before Asynchronous Clock	13	9	–	ns
t_{h1}	–	17	Hold time after External Synchronous Clock, 4PT bypass	7	3	–	ns
t_{h2}	–	18	Hold time after External Synchronous Clock	7	3	–	ns
t_{h3}	–	19	Hold time after Internal Synchronous Clock	11	5	–	ns
t_{h4}	–	20	Hold time after Asynchronous Clock	11	5	–	ns
t_{rw1}	–	21	External Reset Pulse Duration	15	13	–	ns
t_{rw2}	–	22	Asynchronous Reset Pulse Duration	15	13	–	ns
t_{wh1}, t_{wl1}	–	23,24	External Synchronous Clock Pulse Duration, High, Low	10	8	–	ns
t_{wh2}, t_{wl2}	–	25,26	Asynchronous Clock Pulse Duration, High, Low	10	8	–	ns

1. External Parameters are tested and guaranteed.
2. See Timing Technical Note for further details.
3. Unless noted, all parameters use ORP, GRP fanout of four, PTSA, and are measured with 16 outputs switching.
4. Standard 16-bit counter implementation using GRP feedback.
5. Clock to output specifications include a maximum skew of 2 ns.
6. Refer to Switching Test Conditions section.

Switching Characteristics[1, 2, 3] pLSI 1032-50

Using I/O Cell

PARAMETER	TEST[4] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
t_{su5}	–	27	Setup Time before External Synchronous Clock	10	5	–	ns
t_{su6}	–	28	Setup Time before Internal Synchronous Clock	0	-5	–	ns
t_{h5}	–	29	Hold Time after External Synchronous Clock	12	6	–	ns
t_{h6}	–	30	Hold Time after Internal Synchronous Clock	20	15	–	ns
t_{wh3}, t_{wl3}	–	31,32	Clock Pulse Duration, High, Low	10	8	–	ns

1. External Parameters are tested and guaranteed.
2. See Timing Technical Note for further details.
3. Unless noted, all parameters use ORP, GRP fanout of four, PTSA, and are measured with 16 outputs switching.
4. Refer to Switching Test Conditions section.

Architectural Description

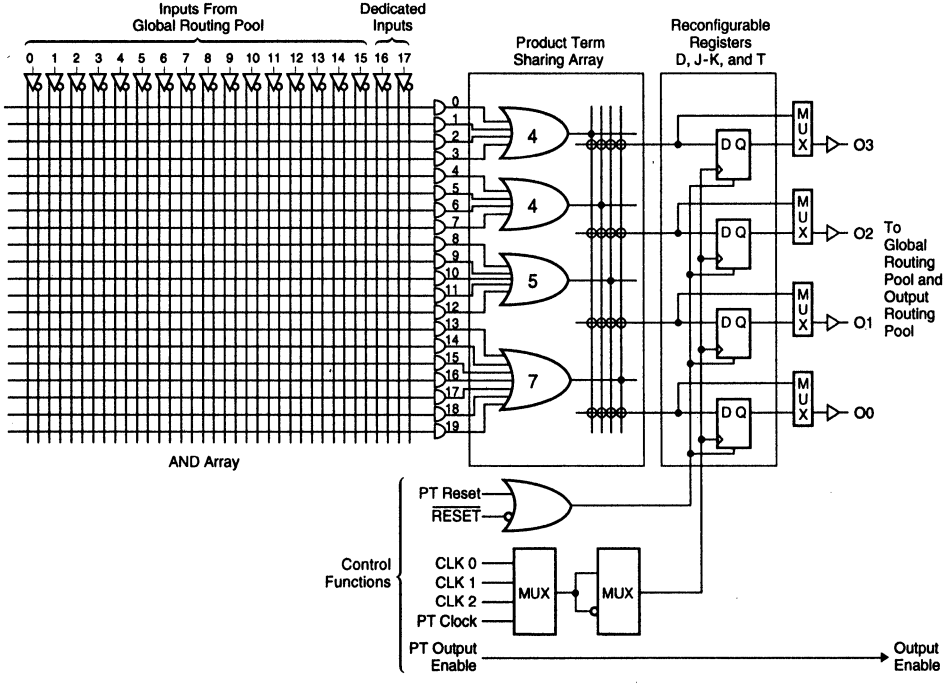
The Generic Logic Block

The Generic Logic Block (GLB) is the standard logic block of the Lattice High Density pLSI Device. This GLB has 18 inputs, four outputs and the logic necessary to implement most standard logic functions. The internal logic of the GLB is divided into four separate sections (see figure 3). The AND Array, the Product Term Sharing Array (PTSA), the Reconfigurable Registers, and the Control Functions. The AND array consists of 20 product terms which can produce the logical sum of any of the 18 GLB inputs. Sixteen of the inputs come from the Global Routing Pool, and are either feedback signals from any of the 32 GLBs or inputs from the external I/O Cells. The two remaining inputs come directly from two dedicated input pins. These signals are available to the product terms in both the logical true and the complemented forms which makes boolean logic reduction easier.

The PTSA takes the 20 product terms and allocates them to the four GLB outputs. There are four OR gates, with four, four, five and seven product terms (see figure 3). The output of any of these gates can be routed to any of the four GLB outputs, and if more product terms are needed, the PTSA can combine them as necessary. If the users' main concern is speed, the PTSA can use a bypass circuit with four product terms to increase the performance of the cell (see figure 4). This can be done to any or all of the four outputs from the GLB.

The Reconfigurable Registers consist of four D-type flip-flops with an Exclusive OR Gate on the input. The Exclusive OR gate in the GLB can be used either as a logic element or to reconfigure the D-type flip-flop to emulate a J-K, or T-type flip-flop (see figure 5). This greatly simplifies the design of counters, comparators and ALU type functions. The registers can be bypassed, if the user needs a combinatorial output. Each register output is brought back into the Global Routing Pool and is also brought to the I/O cells via the Output Routing Pool. Reconfigurable registers are not available when the four product term bypass is used.

Figure 3. GLB: Product Term Sharing Array



Architectural Description

Figure 4. GLB: Four Product Term Bypass

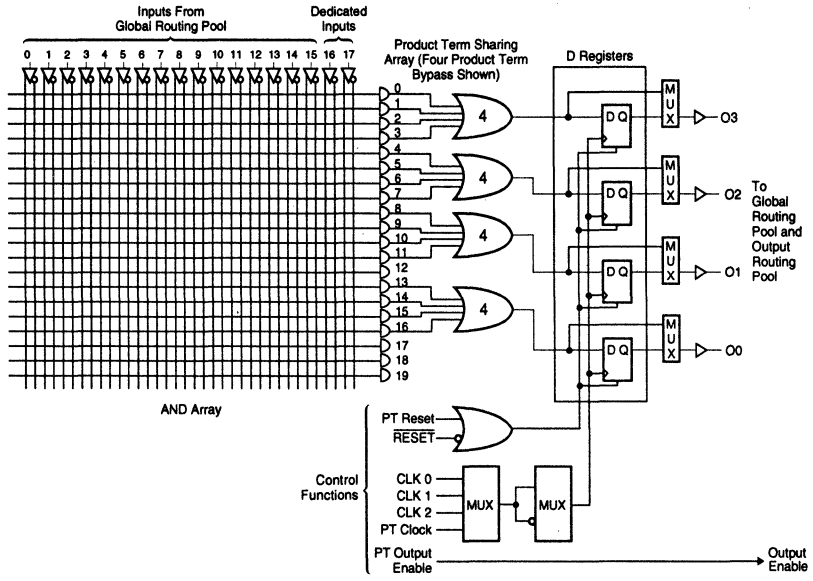
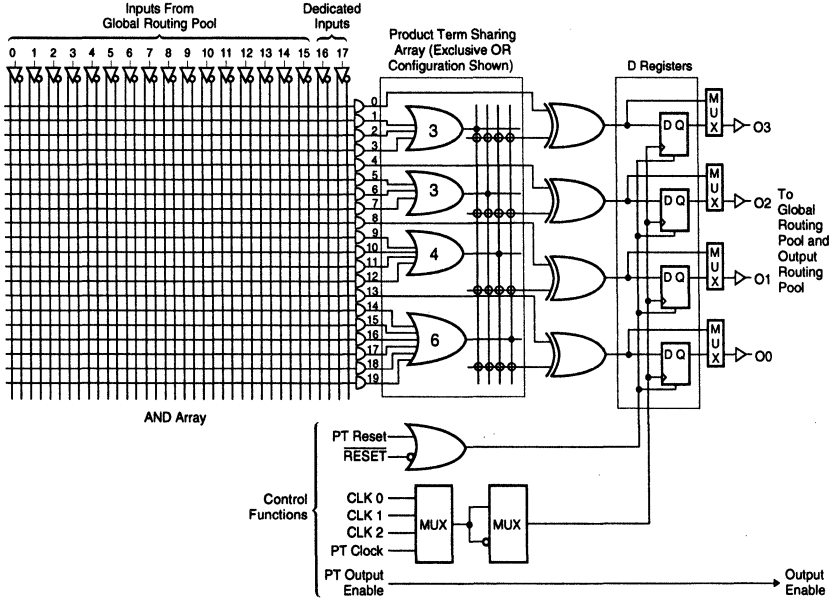


Figure 5. GLB: Exclusive OR Gate



Architectural Description

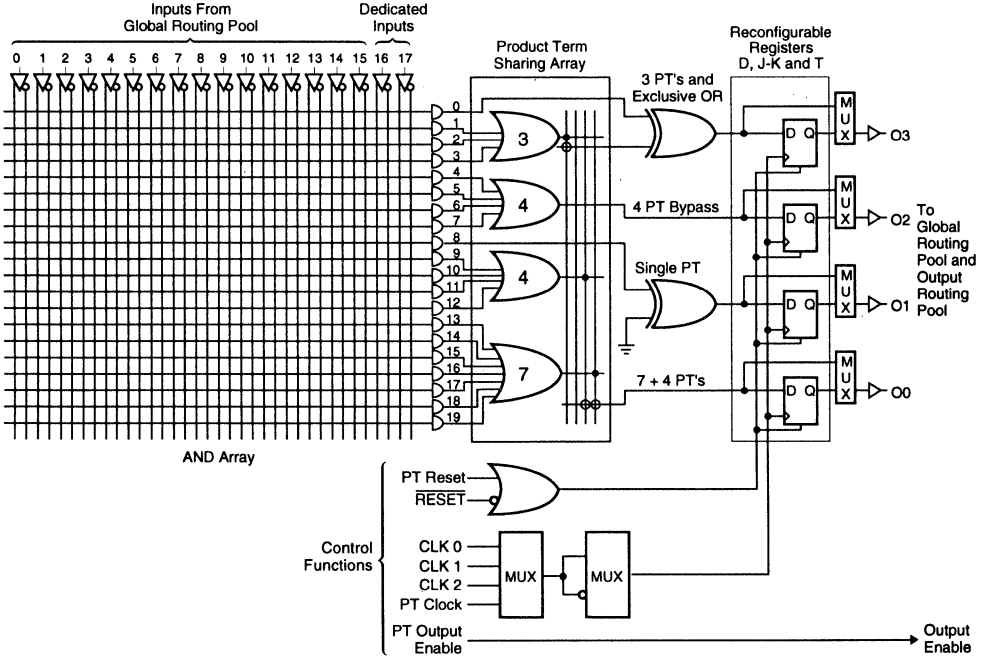
The Generic Logic Block (continued)

The PTSA is flexible enough to allow these features to be used in virtually any combination that the user desires. In the GLB shown in figure 6, Output Three (O3) is configured using the XOR Gate and Output Two (O2) is configured using the four Product Term Bypass. Output One (O1) uses one of the inputs from the five Product Term OR gate while Output Zero (O0) combines the remaining four product terms with all of the product terms from the seven Product Term OR gate for a total of eleven (7+4).

Various signals which control the operation of the GLB are developed in the Control Function section. The clock for the registers can come from any of three sources developed in the Clock Distribution Network (See Clock Distribution Network Section) or from a product term within the GLB. The Reset Signal for the GLB can come from the Global Reset pin or from a product term within the block. The Output Enable for the I/O cells associated with the GLB comes from a product term within the block. Use of a product term for a control function makes that product term unavailable for use as a logic term. Refer to the following table to determine which logic functions are affected.

There are many additional features in a GLB which allow implementation of logic intensive functions. These features are accessible using the hard Macros from the software and require no intervention on the part of the user.

Figure 6. GLB: Various Logical Combinations



Architectural Description

Product Term Sharing Matrix

Product Term #	Standard Configuration Output Number 3 2 1 0	Four Product Term Bypass Output Number 3 2 1 0	Single Product Term Output Number 3 2 1 0	XOR Function Output Number 3 3 2 2 1 1 0 0	Alternate Function
0	■ ■ ■ ■	■	■	■	
1	■ ■ ■ ■	■		■	
2	■ ■ ■ ■	■		■	
3	■ ■ ■ ■	■		■	
4	■ ■ ■ ■	■	■	■	
5	■ ■ ■ ■	■		■	
6	■ ■ ■ ■	■		■	
7	■ ■ ■ ■	■		■	
8	■ ■ ■ ■	■	■	■	
9	■ ■ ■ ■	■		■	
10	■ ■ ■ ■	■		■	
11	■ ■ ■ ■	■		■	
12	■ ■ ■ ■	■		■	■ CLK/Reset
13	■ ■ ■ ■	■	■	■	
14	■ ■ ■ ■	■		■	
15	■ ■ ■ ■	■		■	
16	■ ■ ■ ■	■		■	
17	■ ■ ■ ■			■	
18	■ ■ ■ ■			■	
19	■ ■ ■ ■			■	■ OE/Reset

This matrix shows how each of the product terms are used in the various modes. As an example, Product Term 12 can be used as an input to the five input OR gate in the standard configuration. This OR gate under standard configuration can be routed to any of the four GLB outputs. Product Term 12 is not used in the four product term bypass mode. When GLB output one is used in the Exclusive OR (XOR) mode Product Term 12 becomes one of the inputs to the four input OR Gate. If Product Term 12 is not used in the logic, then it is available for use as either the Asynchronous Clock signal or the GLB Reset signal.

Architectural Description

The Megablock

The pLSI family is structured into multiple "Megablocks". A Megablock consists of 8 GLBs, an Output Routing Pool (ORP) and 16 I/O Cells. Each of these will be explained in detail in the following sections. These elements are coupled together as shown in figure 7. This logic structure is referred to as the Megablock. The various members of the pLSI family are created by combining from two to six Megablocks on a single device.

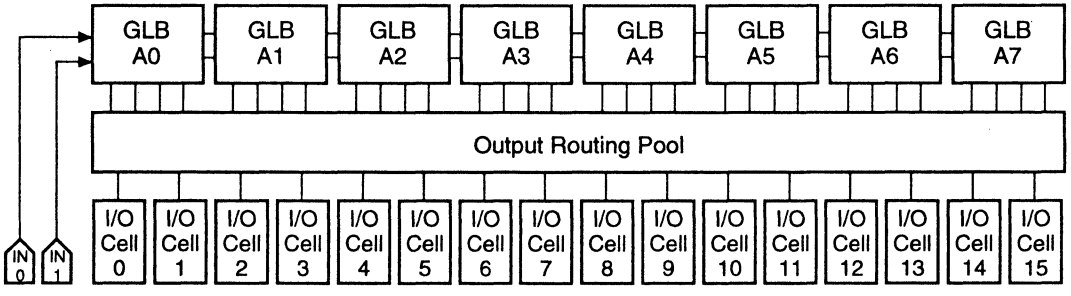
The Megablock shares two sets of resources. The eight GLBs within the Megablock share two dedicated input pins. These dedicated input pins are not available to GLBs in any other Megablock. These pins are dedicated (non-registered) inputs only and are automatically assigned by software. One Output Enable signal is generated within the Megablock and is common to all sixteen of the I/O Cells in the Megablock. The Output Enable signal can be generated using a product term in any of the eight GLBs within the Megablock (see the following section on the Output Enable Multiplexers).

Because of the shared logic within the Megablock, signals which share a common function (counters, busses, etc.) should be grouped within a Megablock. This will allow the user to obtain the best utilization of the logic within the device and eliminate routing bottlenecks.

Megablocks in Each Device

DEVICE	MEGABLOCKS	GLBs	I/O CELLS
pLSI 1016	2	16	32
pLSI 1024	3	24	48
pLSI 1032	4	32	64
pLSI 1048	6	48	96

Figure 7. The Megablock



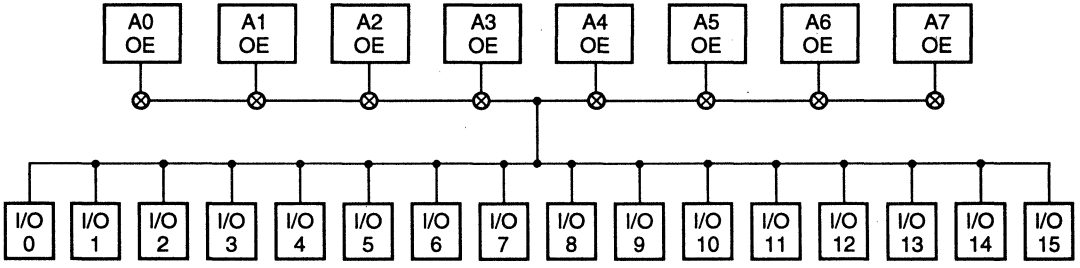
Note: The inputs from the I/O Cell are not shown in this figure, they go directly to the GRP.

Architectural Description

The Output Enable Control

One Output Enable (OE) signal can be generated within each GLB using the OE Product Term 19. One of the eight OE signals from each GLB within a Megablock, is then routed to all of the I/O Cells within the Megablock (see figure 8). This OE signal can simultaneously control all of the 16 I/O cells which are used in 3-state mode. Individual I/O cells also have independent control for permanently enabling or disabling the output buffer (refer to the I/O Cell section). Only one OE signal is allowed per Megablock for 3-state operation. The advantage to this approach is that the OE signal can be generated in any GLB within the Megablock which happens to have an unused OE product term. This frees up the other OE product terms for use as logic.

Figure 8. Output Enable Controls



Architectural Description

The Output Routing Pool

The Output Routing Pool (ORP) routes signals from the Generic Logic Block outputs to I/O Cells configured as outputs or bi-directional pins (see figure 9). The purpose of the ORP is to allow greater flexibility when assigning I/O pins. It also simplifies the job for the routing software which results in a higher degree of utilization.

By examining the programmable switch matrix in figure 9, it can be seen that a GLB output can be connected to one of four I/O Cells. Further flexibility is provided by using the PTSA, (figures 3 through 7) which makes the GLB outputs completely interchangeable. This allows the routing program to freely interchange the outputs to achieve the best routability. This is an automatic process and requires no intervention on the part of the user.

The ORP bypass connections (see figure 10) further increase the flexibility of the device. The ORP bypass connect specific GLB outputs to specific I/O Cells at a faster speed. The bypass path tends to restrict the routability of the device and should only be used for critical signals.

2

Figure 9. Output Routing Pool

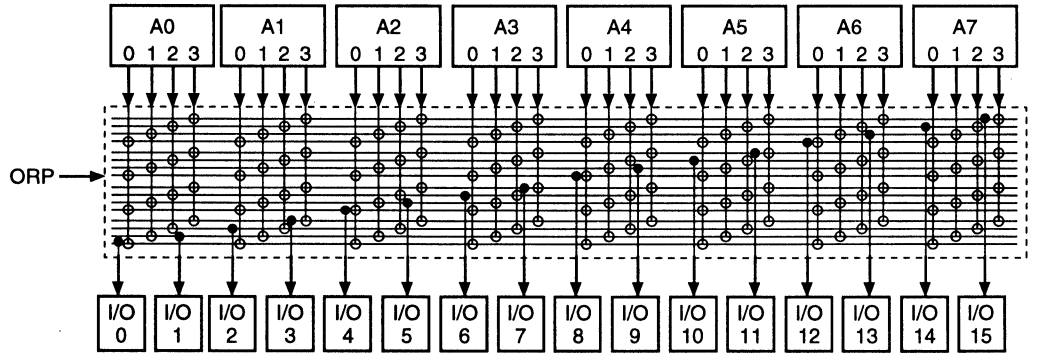
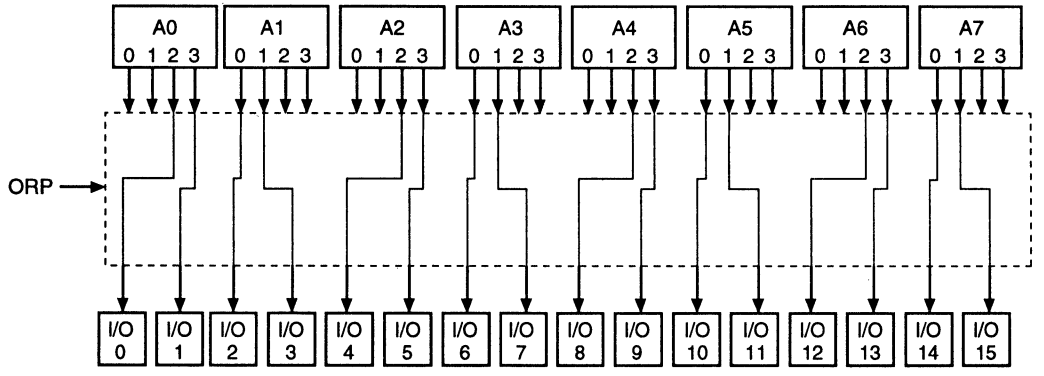


Figure 10. Output Routing Pool showing Bypass



Architectural Description

I/O Cell

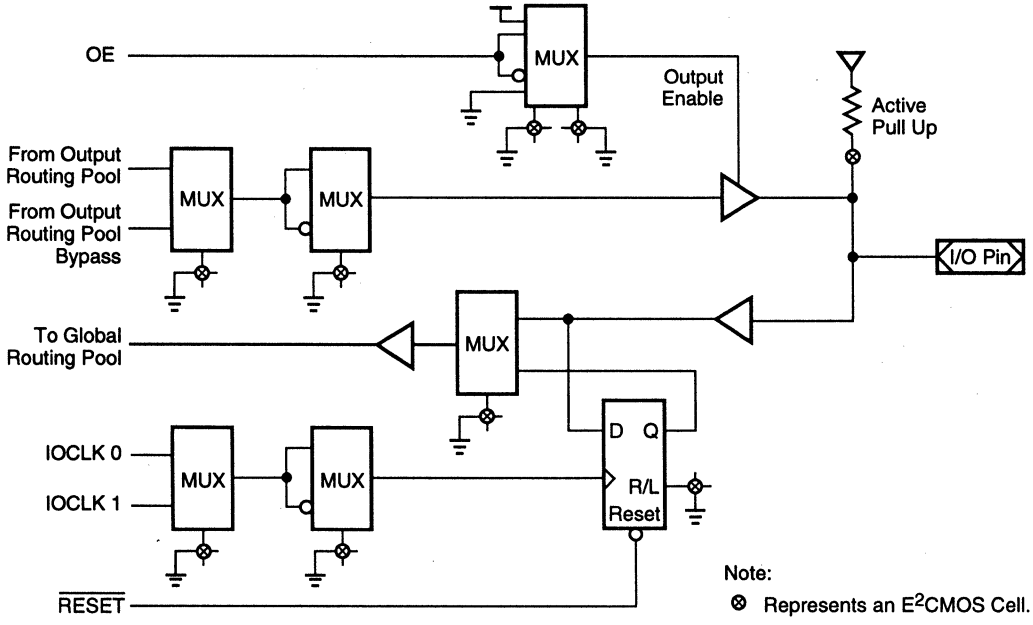
The I/O Cell (see figure 11) is used to route input, output or bidirectional signals connected to the I/O pin. The two logic inputs come from the ORP (see figure 7). One comes from the ORP, and the other comes from the faster ORP bypass. A pair of multiplexers select which signal will be used, and its polarity.

The OE signal comes from the Output Enable. As with the data path, a multiplexer selects the signal polarity. The Output Enable can be set to a logic high (Enabled) when a straight output pin is desired, or logic low (Disabled) when a straight input pin is needed. The Global Reset ($\overline{\text{RESET}}$) signal is driven by the active low chip reset pin. Each I/O Cell can individually select one of the two clock signals (IOCLK 0 or IOCLK 1). These clock signals are generated by the Clock Distribution Network.

Using the multiplexers, the I/O Cell can be configured as an input, an output, a 3-stated output or a bidirectional I/O. The D-type register can be configured as a level sensitive transparent latch or an edge triggered flip-flop to store the in-coming data. Figure 12 illustrates some of the various I/O cell configurations possible.

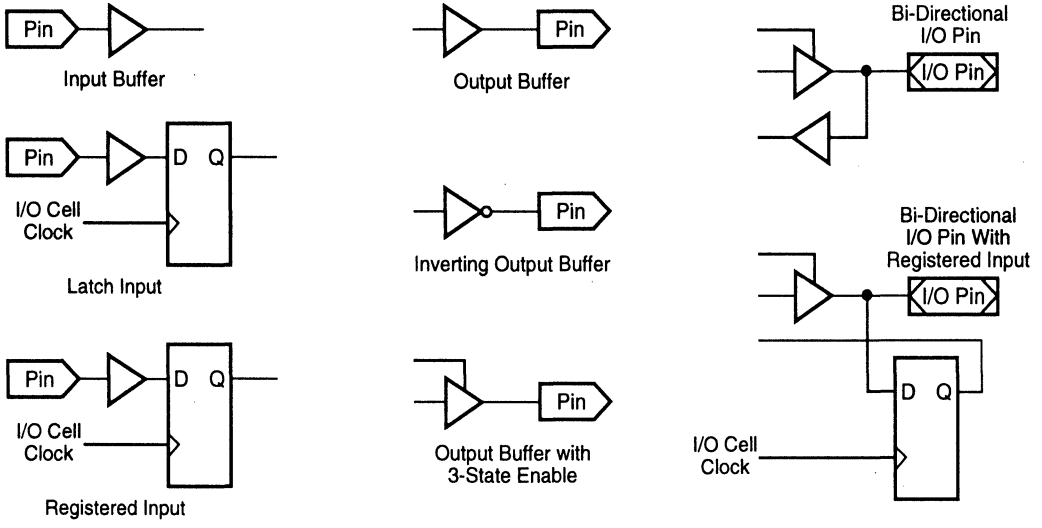
There is an Active Pullup resistor on the I/O pins which is automatically used when the pin is not connected. This prevents the pin from floating and inducing noise into the device or consuming additional power.

Figure 11. I/O Cell Architecture



Architectural Description

Figure 12. Example I/O Cell Configurations



Architectural Description

Clock Distribution Network

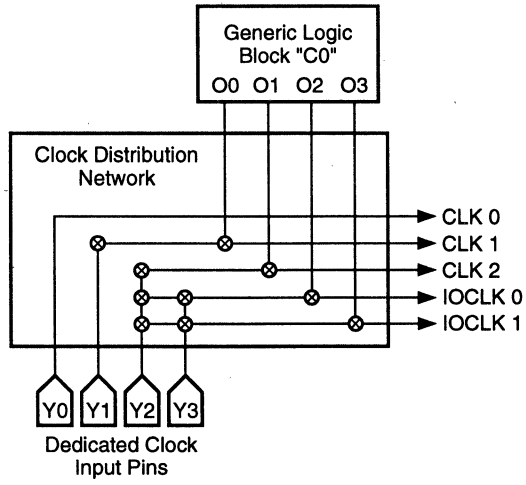
The Clock Distribution Network is shown in figure 13. It generates five global clock signals CLK 0, CLK 1, CLK 2 and IOCLK 0, IOCLK 1. The first three, CLK 0, CLK 1 and CLK 2 are used for clocking all the GLBs in the device. Similarly, IOCLK 0 and IOCLK 1 signals are used for clocking all of the I/O Cells in the device. There are 4 dedicated system clock pins (Y0, Y1, Y2, Y3) which can be directed to any GLB or any I/O Cell using the Clock Distribution Network. The other inputs to the Clock Distribution Network are the 4 outputs of a dedicated clock GLB ("C0" for pLSI 1032). These Clock GLB outputs can be used to create a user-defined internal clocking scheme.

Typically the clock GLB will be clocked using an external main clock pin Y0 connected to global clock signal CLK 0. The outputs of the clock GLB in turn will generate "divide by two" or "divide by four" phases of the CLK 0 which can be connected to CLK 1, CLK 2 or IOCLK 0, IOCLK 1 global clocks.

All GLBs also have the capability of generating their own asynchronous clocks using Product Term 12. CLK 0, CLK 1 and CLK 2 feed to their corresponding inputs on all the GLBs (see figure 3).

The two I/O clocks generated in the Clock Distribution Network IOCLK 0 and IOCLK 1, are brought to all 64 of the I/O cells and the user programs the I/O cell to use one of the two. When the Dedicated Clock Input pins Y2 or Y3 are used as one of the I/O Clocks, O2 or O3 from the Clock GLB (C0) cannot be used.

Figure 13. Clock Distribution Network



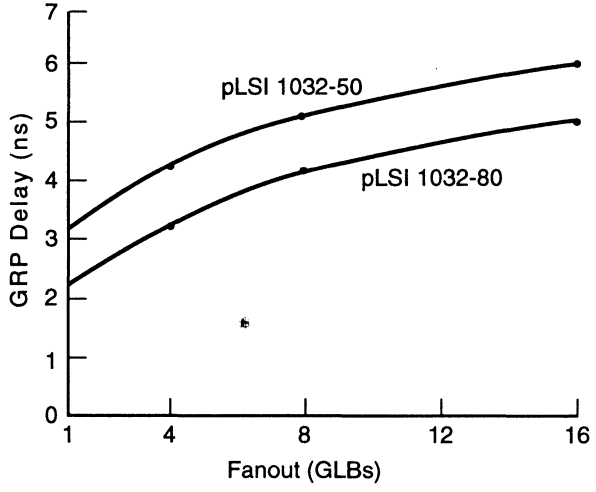
Note: Y3 pin should always be used first as an IOCLK 0 or IOCLK 1 before using Y2 pin.

Architectural Description

Global Routing Pool

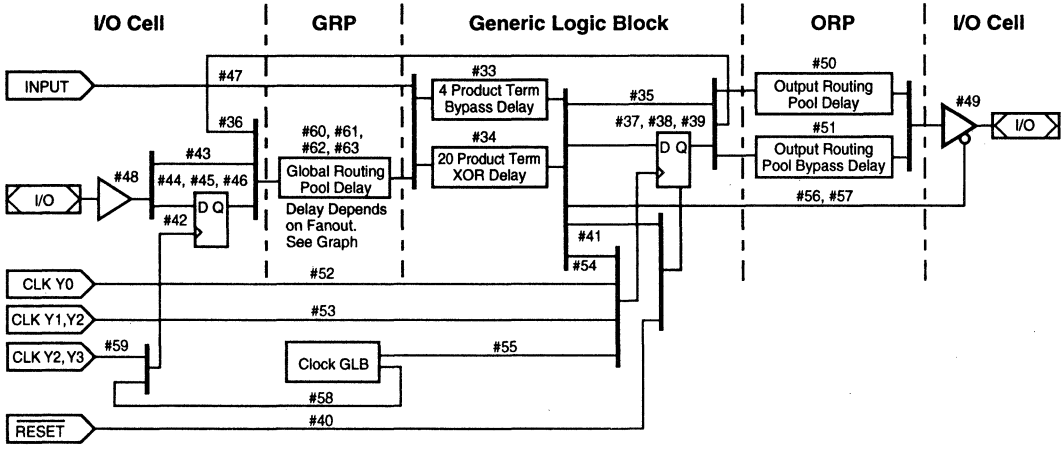
The Global Routing Pool (GRP) is a Lattice proprietary interconnect structure which offers fast predictable speeds with complete connectivity. The GRP allows the outputs from the GLBs or the I/O cell inputs to be connected to the inputs of the GLBs. Any GLB output is available to the input of all other GLBs, and similarly an input from an I/O pin is available as an input to all of the GLBs. Because of the uniform architecture of the pLSI device, the delays through the Global Routing Pool are both consistent and predictable. However, they are slightly affected by fanout. See the fanout delay graph (see figure 14).

Figure 14. GRP Delay vs Fanout



Timing Model

Figure 15. pLSI Timing Model



The task of determining the timing through the device is simple and straightforward. The device timing model is shown in figure 15. To determine the time that it takes for data to propagate through the device, simply determine the path the data is expected to follow, and add the various times. Some examples are shown for the critical timing paths used as Data Sheet parameters. Note that the Internal timing parameters are given for reference only, and are not tested. (External timing parameters are tested and guaranteed on every device).

t_{pd1}[1]	=	t_{ib}	+	t_{iobp}	+	t_{grp4}	+	t_{4pt}	+	t_{gbp}	+	t_{mxbp}	+	t_{ob}		
#1	=	#48	+	#43	+	#60	+	#33	+	#35	+	#51	+	#49		
15 ns	=	2	+	0	+	3	+	6	+	0	+	0	+	4		
t_{pd2}[1]	=	t_{ib}	+	t_{iobp}	+	t_{grp4}	+	t_{xor20}	+	t_{gbp}	+	t_{mx}	+	t_{ob}		
#2	=	#48	+	#43	+	#60	+	#34	+	#35	+	#50	+	#49		
20 ns	=	2	+	0	+	3	+	7.5	+	0	+	1	+	4		
f_{max}[1]	=	t_{gco}	+	t_{gfb}	+	t_{grp4}	+	t_{xor20}	+	t_{gsu}						
#11	=	#37	+	#36	+	#60	+	#34	+	#38						
1 / 14ns	=	2	+	0	+	3	+	7.5	+	0						
t_{su2}[1]	=	t_{ib}	+	t_{iobp}	+	t_{grp4}	+	t_{xor20}	-	t_{gy0}	-	t_{gsu}				
#14	=	#48	+	#43	+	#60	+	#34	-	#52	-	#38				
12 ns	=	2	+	0	+	3	+	7.5	-	4	-	0				
t_{co2}[1]	=	t_{gy0}	+	t_{gco}	+	t_{mx}	+	t_{ob}								
#4	=	#52	+	#37	+	#50	+	#49								
13 ns	=	4	+	2	+	1	+	4								

NOTE:
1. The internal delays are rounded and do not necessarily add up to the tested external delays.

Switching Characteristics — pLSI 1032-80

Internal Timing Model Parameters are not tested and are for reference only

Generic Logic Block (GLB)[1]

PARAMETER	#	DESCRIPTION	MIN.	MAX	UNITS
t_{4pt}	33	4 Product Term Delay	–	6	ns
t_{xor20}	34	20 Product Term Delay	–	7.5	ns
t_{gbp}	35	GLB Register By-Pass Delay	–	0	ns
t_{gfb}	36	GLB Feedback Delay	–	0	ns
t_{gco}	37	GLB Clock to Output Delay	–	2	ns
t_{gsu}	38	GLB Setup Time before Clock	0	–	ns
t_{gh}	39	GLB Hold Time after Clock	2	–	ns

Reset Delays

PARAMETER	#	DESCRIPTION	MIN.	MAX.	UNITS
t_{ggr}	40	GLB Global Reset Delay	–	12	ns
t_{gar}	41	GLB Asynchronous Reset Delay	–	9	ns

Input/Output Cell (I/O Cell)

PARAMETER	#	DESCRIPTION	MIN.	MAX.	UNITS
t_{lat}	42	I/O Cell Latch Delay	–	1	ns
t_{iobp}	43	I/O Cell Register Latch By-Pass Delay	–	0	ns
t_{iosu}	44	I/O Cell Setup Time before Clock/LE	0	–	ns
t_{ioh}	45	I/O Cell Hold Time after Clock/LE	1	–	ns
t_{ioco}	46	I/O Cell Clock/LE to Output Delay	–	1	ns

Input and Output Delays

PARAMETER	#	DESCRIPTION	MIN.	MAX	UNITS
t_{din}	47	Dedicated Input Buffer Delay	–	6	ns
t_{ib}	48	Input Delay for I/O Buffer	–	2	ns
t_{ob}	49	Output Buffer Delay	–	4	ns
t_{mx}	50	Output ORP Delay	–	1	ns
t_{mxbp}	51	Output ORP Bypass Delay	–	0	ns

(Note: Parameter Numbers refer to the timing paths used in the Timing Model Diagram.)

Switching Characteristics — pLSI 1032-80

Internal AC Characteristics and Conditions are not tested and are for reference only

Clock and Enable Delays, GLB and I/O Cell

PARAMETER	#	DESCRIPTION	MIN.	MAX	UNITS
tgy0	52	Clock Delay, Y0 to GLB	–	4	ns
tgy1/2	53	Clock Delay, Y1 or Y2 to GLB	–	4	ns
tgpt	54	Clock Delay, PT Clk to GLB	–	5	ns
tgcp	55	Clock Delay, Clk GLB to GLB	–	4	ns
tgen	56	Enable Delay, GLB to I/O Cell	–	7	ns
tgdis	57	Disable Delay, GLB to I/O Cell	–	7	ns
tiocp	58	Clock Delay, Clk GLB to I/O Cell	–	4	ns
tioy2/3	59	Clock Delay, Y2 or Y3 to I/O Cell	–	4	ns

GRP Delays

PARAMETER	#	DESCRIPTION	MIN.	MAX	UNITS
tgrp4	60	GRP Delay, Fanout 4	–	3	ns
tgrp8	61	GRP Delay, Fanout 8	–	4	ns
tgrp16	62	GRP Delay, Fanout 16	–	5	ns
tgrp32	63	GRP Delay, Fanout 32	–	8	ns

(Note: Parameter Numbers refer to the timing paths used in the Timing Model Diagram.)

Switching Characteristics — pLSI 1032-50

Internal Timing Model Parameters are not tested and are for reference only

Generic Logic Block (GLB)[1]

PARAMETER	#	DESCRIPTION	MIN.	MAX	UNITS
t_{4pt}	33	4 Product Term Delay	–	7	ns
t_{xor20}	34	20 Product Term Delay	–	10	ns
t_{gbp}	35	GLB Register By-Pass Delay	–	0	ns
t_{gfb}	36	GLB Feedback Delay	–	0	ns
t_{gco}	37	GLB Clock to Output Delay	–	3	ns
t_{gsu}	38	GLB Setup Time before Clock	2	–	ns
t_{gh}	39	GLB Hold Time after Clock	4	–	ns

Reset Delays

PARAMETER	#	DESCRIPTION	MIN.	MAX.	UNITS
t_{ggr}	40	GLB Global Reset Delay	–	14	ns
t_{gar}	41	GLB Asynchronous Reset Delay	–	10	ns

Input/Output Cell (I/O Cell)

PARAMETER	#	DESCRIPTION	MIN.	MAX.	UNITS
t_{lat}	42	I/O Cell Latch Delay	–	2	ns
t_{iobp}	43	I/O Cell Register Latch By-Pass Delay	–	0	ns
t_{iosu}	44	I/O Cell Setup Time before Clock/LE	0	–	ns
t_{ioh}	45	I/O Cell Hold Time after Clock/LE	2	–	ns
t_{ioco}	46	I/O Cell Clock/LE to Output Delay	–	2	ns

Input and Output Delays

PARAMETER	#	DESCRIPTION	MIN.	MAX	UNITS
t_{din}	47	Dedicated Input Buffer Delay	–	7	ns
t_{ib}	48	Input Delay for I/O Buffer	–	3	ns
t_{ob}	49	Output Buffer Delay	–	5	ns
t_{mx}	50	Output ORP Delay	–	2	ns
t_{mxbp}	51	Output ORP Bypass Delay	–	0	ns

(Note: Parameter Numbers refer to the timing paths used in the Timing Model Diagram.)

Switching Characteristics — pLSI 1032-50

Internal AC Characteristics and Conditions are not tested and are for reference only

Clock and Enable Delays, GLB and I/O Cell

PARAMETER	#	DESCRIPTION	MIN.	MAX	UNITS
tgy0	52	Clock Delay, Y0 to GLB	–	6	ns
tgy1/2	53	Clock Delay, Y1 or Y2 to GLB	–	6	ns
tgpt	54	Clock Delay, PT Clk to GLB	–	7	ns
tgcp	55	Clock Delay, Clk GLB to GLB	–	5	ns
tgen	56	Enable Delay, GLB to I/O Cell	–	9	ns
tgdis	57	Disable Delay, GLB to I/O Cell	–	9	ns
tiocp	58	Clock Delay, Clk GLB to I/O Cell	–	5	ns
tioy2/3	59	Clock Delay, Y2 or Y3 to I/O Cell	–	5	ns

GRP Delays

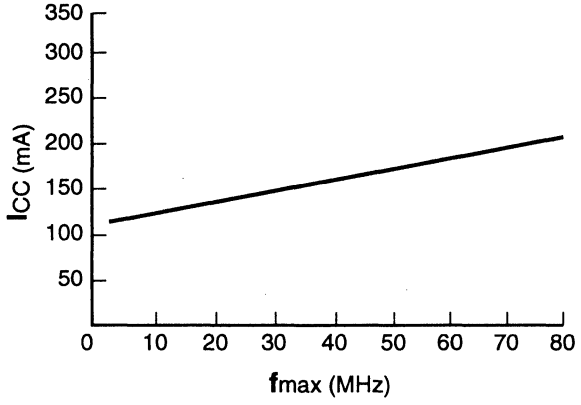
PARAMETER	#	DESCRIPTION	MIN.	MAX	UNITS
tgrp4	60	GRP Delay, Fanout 4	–	4	ns
tgrp8	61	GRP Delay, Fanout 8	–	6	ns
tgrp16	62	GRP Delay, Fanout 16	–	7	ns
tgrp32	63	GRP Delay, Fanout 32	–	10	ns

(Note: Parameter Numbers refer to the timing paths used in the Timing Model Diagram.)

Power Consumption

Power consumption in the pLSI 1032 device depends on two primary factors: the speed at which the device is operating and the number of product terms used. Figure 16 shows the relationship between power and operating speed.

Figure 16. Typical Device Power Consumption vs fmax



Configuration of Eight 16-bit Counters

Security Cell

A security cell is provided in the pLSI devices to prevent unauthorized copying of the array patterns. Once programmed, this cell prevents further read access to the functional bits in the device. This cell can only be erased by reprogramming the device, so the original configuration can never be examined once this cell is programmed.

Device Programming

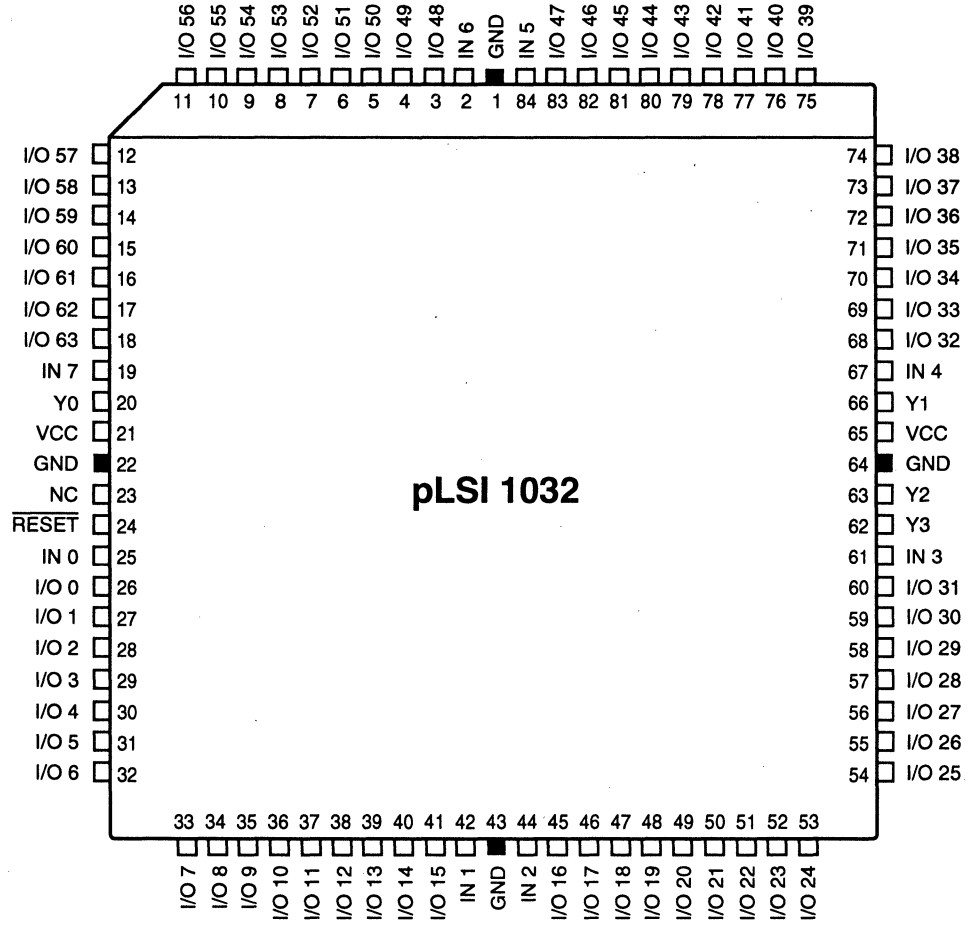
pLSI devices are programmed using a Lattice-approved Device Programmer, available from a number of third party manufacturers. Complete programming of the device takes only a few seconds. Erasing of the device is automatic and is completely transparent to the user. In-system programming is available with ispLSI devices using Lattice programming algorithms.

Latch-up Protection

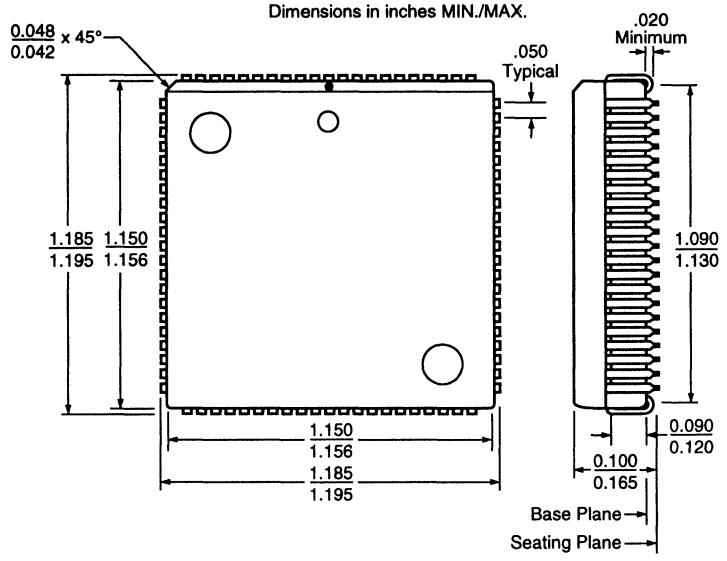
pLSI devices are designed with an on-board charge pump to negatively bias the substrate. The negative bias is of sufficient magnitude to prevent input undershoots from causing the internal circuitry to latchup. Additionally, outputs are designed with n-channel pull-ups instead of the traditional p-channel pull-ups to eliminate any possibility of SCR induced latching.

Pin Configuration

pLSI 1032 PLCC Pinout Diagram



84-Pin PLCC


Dimensions in inches MIN./MAX.
0.048 / 0.042 x 45°
.050 Typical
.020 Minimum
1.185 / 1.195
1.150 / 1.156
1.150 / 1.156
1.185 / 1.195
1.090 / 1.130
0.090 / 0.120
0.100 / 0.165
Base Plane
Seating Plane

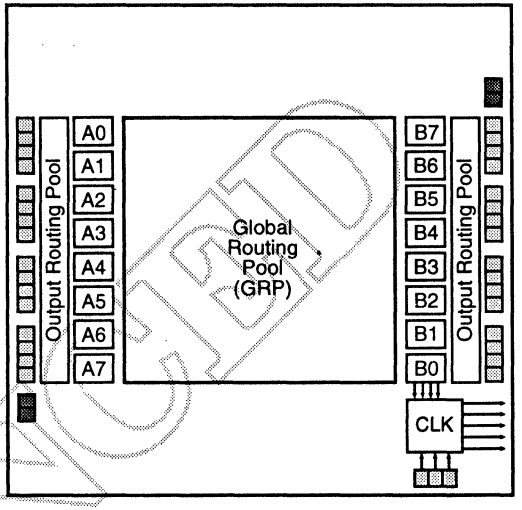
pLSI™ 1016

programmable Large Scale Integration

Features

- **PROGRAMMABLE HIGH DENSITY LOGIC**
  - — Member of Lattice's pLSI Family
  - — High Speed Global Interconnects
  - — 32 I/O Pins, Four Dedicated Inputs
  - — 96 Registers
  - — Wide Input Gating for Fast Counters, State Machines, Address Decoders, etc.
  - — Small Logic Block Size for Random Logic
  - — Security Cell Prevents Unauthorized Copying
- **HIGH PERFORMANCE E²CMOS® TECHNOLOGY**
  - — f_{max} = 80 MHz Maximum Operating Frequency
  - — t_{pd} = 15 ns Propagation Delay
  - — TTL Compatible Inputs and Outputs
  - — Electrically Erasable and Re-Programmable
  - — 100% Tested
- **COMBINES EASE OF USE AND THE FAST SYSTEM SPEED OF PLDs WITH THE DENSITY AND FLEXIBILITY OF FIELD PROGRAMMABLE GATE ARRAYS**
  - — Complete Programmable Device can Combine Glue Logic and Structured Designs
  - — 100% Routable at 80% Utilization
  - — Three Dedicated Clock Input Pins
  - — Synchronous and Asynchronous Clocks
  - — Flexible Pin Placement
  - — Optimized Global Routing Pool Allows Global Interconnectivity
- **pLSI/ispLSI™ DEVELOPMENT SYSTEM (pDS™)**
  - — Boolean Logic Compiler
  - — Automatic Place and Route
  - — Manual Partitioning
  - — PC Platform
  - — Easy to Use Windows Interface
- **ADVANCED pLSI/ispLSI DEVELOPMENT SYSTEM**
  - — Industry Standard, Third Party Design Environments
  - — Schematic Capture
  - — Fully Automatic Partitioning
  - — Automatic Place and Route
  - — Comprehensive Logic and Timing Simulation
  - — PC and Workstation Platforms

Functional Block Diagram



Description

The Lattice pLSI 1016 is a High Density Programmable Logic Device which contains 96 Registers, 32 Universal I/O pins, four Dedicated Input Pins, three Dedicated Clock Input Pins and a Global Routing Pool (GRP). The GRP provides complete interconnectivity between all of these elements.

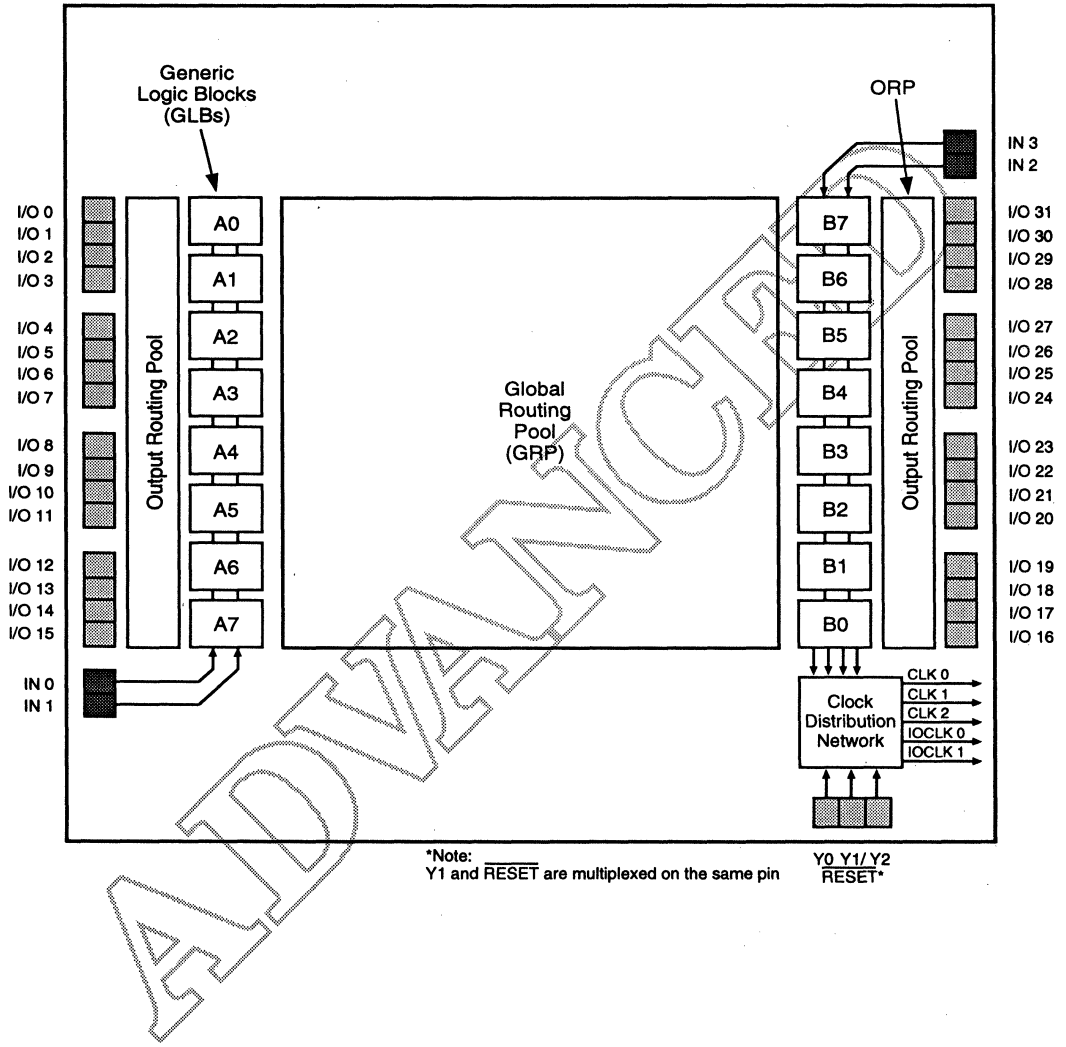
The basic unit of logic on the pLSI 1016 device is the Generic Logic Block (GLB). The GLBs are labeled A0, A1 ..B7, (see figure 1). There are a total of 16 GLBs in the pLSI 1016 device. Each GLB has 18 inputs, a programmable AND/OR/Exclusive OR array, and four outputs which can be configured to be either combinatorial or registered. Inputs to the GLB come from the GRP. All of the GLB outputs are brought back into the GRP so that they can be connected to the inputs of any other GLB on the device.

LATTICE SEMICONDUCTOR CORP., 5555 Northeast Moore Ct., Hillsboro, Oregon 97124, U.S.A.
Tel. 1-800-LATTICE (528-8423); FAX (503) 681-3037

January 1992. Rev. A

Functional Block Diagram

Figure 1. pLSI 1016



Description (continued)

The device also has 32 I/O Cells, each of which is directly connected to an I/O pin. Each I/O cell can be individually programmed to be a combinatorial input, registered input, latched input, output or bi-directional I/O pin with 3-state. The signal levels are TTL compatible voltages and the output drivers can source 4 mA or sink 8 mA.

The 32 I/O Cells are grouped into two sets of 16 each as shown in figure 1. Each of these I/O groups is associated with a logic Megablock through the use of the Output Routing Pool (ORP) and shares a common Output Enable (OE) signal.

Eight GLBs, 16 I/O Cells and one ORP are connected together to make a logic Megablock. The Megablock is defined by the resources that it shares. The outputs of the eight GLBs are connected to a set of 16 universal I/O cells by the ORP. The pLSI 1016 Device contains two of these Megablocks.

The GRP has as its inputs the outputs from all of the GLBs and all of the inputs from the bi-directional I/O cells. All of these signals are made available to the inputs of the GLBs. Delays through the GRP have been equalized to minimize timing skew and logic glitching.

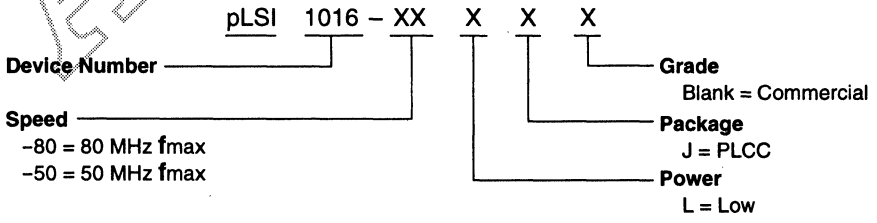
Clocks in the pLSI 1016 device are selected using the Clock Distribution Network. Three dedicated clock pins (Y0 to Y2) are brought into the distribution network, and five outputs (CLK 0 to CLK 2 and IOCLK 0, IOCLK 1) are provided to route clocks to the GLBs and I/O cells. The Clock Distribution Network can also be driven from a special GLB (B0 on the pLSI 1016 device). The logic of this GLB allows the user to create an internal clock from a combination of internal signals within the device.

The pLSI 1016 device is part of Lattice's programmable Large Scale Integration (pLSI) family. This family contains a range of devices from the pLSI 1016, with 96 registers, to the pLSI 1048 with 288 registers. The pLSI Family Product Selector Guide below lists key attributes of the devices along with the number of resources available.

pLSI Family Product Selector Guide

DEVICE	pLSI 1016	pLSI 1024	pLSI 1032	pLSI 1048
GLBs	16	24	32	48
Registers	96	144	192	288
I/O Pins	32	48	64	96
Dedicated Inputs	4	6	8	10
Pin Count	44	68	84	120

Ordering Information



Pin Description

Name	PLCC Pin Numbers	Description
I/O 0 - I/O 3 I/O 4 - I/O 7 I/O 8 - I/O 11 I/O 12 - I/O 15 I/O 16 - I/O 19 I/O 20 - I/O 23 I/O 24 - I/O 27 I/O 28 - I/O 31	15, 16, 17, 18, 19, 20, 21, 22, 25, 26, 27, 28, 29, 30, 31, 32, 37, 38, 39, 40, 41, 42, 43, 44, 3, 4, 5, 6, 7, 8, 9, 10	Input/Output Pins - These are the general purpose I/O pins used by the logic array.
IN 0 - IN 3	14, 24, 36, 2	Dedicated input pins to the device.
Y0	11	Dedicated Clock input. This clock input is connected to one of the clock inputs of all of the GLBs on the device.
Y1/$\overline{\text{RESET}}$	35	This pin performs two functions: – Dedicated clock input. This clock input is brought into the Clock Distribution Network, and can optionally be routed to any GLB and/or I/O Cell on the device. – Active Low (0) Reset pin which resets all of the GLB and I/O registers in the device.
Y2	33	Dedicated Clock input. This clock input is brought into the clock distribution network, and can optionally be routed to any I/O Cell in the device. This is a factory test pin and it should be left floating or tied to V_{CC}.
NC	13	This is a factory test pin and it should be left floating or tied to V_{CC}.
GND VCC	1, 23 12, 34	Ground (GND) V_{CC}

Absolute Maximum Ratings [1]

Supply Voltage V_{CC} . . . . . -0.5 to +7.0V

Input Voltage Applied. . . . . .-2.5 to V_{CC} +1.0V

Off-State Output Voltage Applied . . . . . .-2.5 to V_{CC} +1.0V

Storage Temperature . . . . . .-65 to 125°C

Ambient Temp. with Power Applied . . . . . .-55 to 125°C

1. Stresses above those listed under the "Absolute Maximum Ratings" may cause permanent damage to the device. Functional operation of the device at these or at any other conditions above those indicated in the operational sections of this specification is not implied (while programming, follow the programming specifications).

DC Recommended Operating Condition

SYMBOL	PARAMETER	MIN.	MAX.	UNITS
T_A	Ambient Temperature	0	70	°C
V_{CC}	Supply Voltage	4.75	5.25	V
V_{IL}	Input Low Voltage	0	0.8	V
V_{IH}	Input High Voltage	2.0	V_{CC}	V

Capacitance (T_A=25°C,f=1.0 MHz)

SYMBOL	PARAMETER	MAXIMUM[1]	UNITS	TEST CONDITIONS
C_1	Input Capacitance	8	pf	V_{CC}=5.0V, V_{IN}=2.0V
C_2	I/O, Y Capacitance	10	pf	V_{CC}=5.0V, VI/O, Y=2.0V

1. Guaranteed but not 100% tested.

Data Retention Specifications

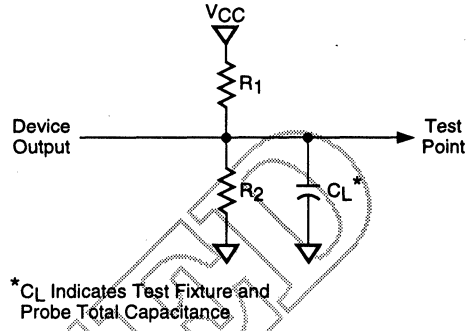
PARAMETER	MINIMUM	MAXIMUM	UNITS
Data Retention	20	–	YEARS
Erase/Reprogram Cycles	–	100	CYCLES

Switching Test Conditions

Input Pulse Levels	GND to 3.0V
Input Rise and Fall Times	3ns 10% to 90%
Input Timing Reference Levels	1.5V
Output Timing Reference Levels	1.5V
Output Load	See Figure 2

3-state levels are measured 0.5V from steady-state active level.

Figure 2. Test Load



Output Load Conditions (see figure 2)

Test Condition		R1	R2	CL
1		470Ω	390Ω	35pF
2	Active High	∞	390Ω	35pF
	Active Low	470Ω	390Ω	35pF
3	Active High to Z at V_{OH} - 0.5V	∞	390Ω	5pF
	Active Low to Z at V_{OL} + 0.5V	470Ω	390Ω	5pF

DC Electrical Characteristics

Over Recommended Operating Conditions

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNITS
V_{OL}	Output Low Voltage	I_{OL} =8 mA.	–	–	0.4	V
V_{OH}	Output High Voltage	I_{OH} =-4 mA.	2.4	–	–	V
I_{IL}	Input or I/O Low Leakage Current	$0V \le V_{IN} \le V_{IL}$ (MAX.)	–	–	-10	µA
I_{IH}	Input or I/O High Leakage Current	$V_{IH} \le V_{IN} \le V_{CC}$	–	–	10	µA
I_{OS}[1]	Output Short Circuit Current	V_{CC} = 5V, V_{OUT} = 0.5V	-60	–	-200	mA
I_{CC}[2]	Operating Power Supply Current	V_{IL} = 0.5V, V_{IH} = 3.0V F_{TOGGLE} = 20 MHz	–	–	–	mA

1. One output at a time for a maximum duration of one second. (Vout = 0.5V)
2. Measured at a frequency of 20 MHz using four 16-bit counters.

External Switching Characteristics[1, 2, 3] pLSI 1016-80

Over Recommended Operating Conditions

PARAMETER	TEST[6] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
t_{pd1}	1	1	Data Propagation Delay, 4PT bypass, ORP bypass	–	12	15	ns
t_{pd2}	1	2	Data Propagation Delay	–	15	20	ns
t_{co1}[5]	1	3	External Clock to Output Delay, ORP bypass	–	8	11	ns
t_{co2}[5]	1	4	External Clock to Output Delay	–	9	14	ns
t_{co3}	1	5	Internal Synch. Clock to Output Delay	–	15	20	ns
t_{co4}	1	6	Asynchronous Clock to Output Delay	–	13	20	ns
t_{r1}	1	7	External Pin Reset to Output Delay	–	13	20	ns
t_{r2}	1	8	Asynchronous PT Reset to Output Delay	–	15	22	ns
t_{en}	2	9	Input to Output Enable	–	13	20	ns
t_{dis}	3	10	Input to Output Disable	–	13	20	ns

External AC Recommended Operating Conditions[1, 2, 3] pLSI 1016-80

Over Recommended Operating Conditions

PARAMETER	TEST[6] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
f_{max}[4]	1	11	Clock Frequency with Internal Feedback	–	100	80	MHz
f_{max} (External)	1	12	Clock Frequency with External Feedback	–	70	50	MHz
t_{su1}	–	13	Setup Time before External Synch. Clock, 4PT bypass	9	6	–	ns
t_{su2}	–	14	Setup Time before External Synch Clock	12	8	–	ns
t_{su3}	–	15	Setup Time before Internal Synch. Clock	9	3	–	ns
t_{su4}	–	16	Setup Time before Asynchronous Clock	9	4	–	ns
t_{h1}	–	17	Hold time after External Synchronous Clock, 4PT bypass	2	-1	–	ns
t_{h2}	–	18	Hold time after External Synchronous Clock	2	-1	–	ns
t_{h3}	–	19	Hold time after Internal Synchronous Clock	8	2	–	ns
t_{h4}	–	20	Hold time after Asynchronous Clock	8	1	–	ns
t_{rw1}	–	21	External Reset Pulse Duration	10	8	–	ns
t_{rw2}	–	22	Asynchronous Reset Pulse Duration	10	8	–	ns
t_{wh1}, t_{wl1}	–	23,24	External Synchronous Clock Pulse Duration, High, Low	6	5	–	ns
t_{wh2}, t_{wl2}	–	25,26	Asynchronous Clock Pulse Duration, High, Low	6	5	–	ns

1. External Parameters are tested and guaranteed.
2. See Timing Technical Note for further details.
3. Unless noted, all parameters use ORP, GRP fanout of four, PTSA, and are measured with 16 outputs switching.
4. Standard 16-bit counter implementation using GRP feedback.
5. Clock to output specifications include a maximum skew of 2 ns.
6. Refer to Switching Test Conditions section.

Switching Characteristics[1, 2, 3] — pLSI 1016-80

Using I/O Cell

PARAMETER	TEST[4] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
t_{su5}	–	27	Setup Time before External Synchronous Clock	5	0	–	ns
t_{su6}	–	28	Setup Time before Internal Synchronous Clock	0	-3	–	ns
t_{h5}	–	29	Hold Time after External Synchronous Clock	8	4	–	ns
t_{h6}	–	30	Hold Time after Internal Synchronous Clock	15	11	–	ns
t_{wh3}, t_{wl3}	–	31,32	Clock Pulse Duration, High, Low	6	5	–	ns

1. External Parameters are tested and guaranteed.
2. See Timing Technical Note for further details.
3. Unless noted, all parameters use ORP, GRP fanout of four, PTSA, and are measured with 16 outputs switching.
4. Refer to Switching Test Conditions section.

External Switching Characteristics[1, 2, 3] pLSI 1016-50

Over Recommended Operating Conditions

PARAMETER	TEST[6] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
t_{pd1}	1	1	Data Propagation Delay, 4PT bypass, ORP bypass	–	16	20	ns
t_{pd2}	1	2	Data Propagation Delay, ORP	–	19	25	ns
t_{co1}[5]	1	3	External Clock to Output Delay, ORP bypass	–	12	16	ns
t_{co2}[5]	1	4	External Clock to Output Delay	–	15	20	ns
t_{co3}	1	5	Internal Synch. Clock to Output Delay	–	21	28	ns
t_{co4}	1	6	Asynchronous Clock to Output Delay	–	21	28	ns
t_{r1}	–	7	External Pin Reset to Output Delay	–	21	28	ns
t_{r2}	–	8	Asynchronous PT Reset to Output Delay	–	24	30	ns
t_{en}	2	9	Input to Output Enable	–	21	28	ns
t_{dis}	3	10	Input to Output Disable	–	21	28	ns

External AC Recommended Operating Conditions[1, 2, 3] pLSI 1016-50

Over Recommended Operating Conditions

PARAMETER	TEST[6] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
f_{max}[4]	1	11	Clock Frequency with Internal Feedback	–	70	50	MHz
f_{max} (External)	1	12	Clock Frequency with External Feedback	–	45	33	MHz
t_{su1}	–	13	Setup Time before External Synch. Clock, 4PT bypass	14	10	–	ns
t_{su2}	–	14	Setup Time before External Synch Clock	17	13	–	ns
t_{su3}	–	15	Setup Time before Internal Synch. Clock	13	9	–	ns
t_{su4}	–	16	Setup Time before Asynchronous Clock	13	9	–	ns
t_{h1}	–	17	Hold time after External Synchronous Clock, 4PT bypass	7	3	–	ns
t_{h2}	–	18	Hold time after External Synchronous Clock	7	3	–	ns
t_{h3}	–	19	Hold time after Internal Synchronous Clock	11	5	–	ns
t_{h4}	–	20	Hold time after Asynchronous Clock	11	5	–	ns
t_{rw1}	–	21	External Reset Pulse Duration	15	13	–	ns
t_{rw2}	–	22	Asynchronous Reset Pulse Duration	15	13	–	ns
t_{wh1}, t_{wl1}	–	23,24	External Synchronous Clock Pulse Duration, High, Low	10	8	–	ns
t_{wh2}, t_{wl2}	–	25,26	Asynchronous Clock Pulse Duration, High, Low	10	8	–	ns

1. External Parameters are tested and guaranteed.
2. See Timing Technical Note for further details.
3. Unless noted, all parameters use ORP, GRP fanout of four, PTSA, and are measured with 16 outputs switching.
4. Standard 16-bit counter implementation using GRP feedback.
5. Clock to output specifications include a maximum skew of 2 ns.
6. Refer to Switching Test Conditions section.

Switching Characteristics[1, 2, 3] pLSI 1016-50

Using I/O Cell

PARAMETER	TEST[4] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
t_{su5}	–	27	Setup Time before External Synchronous Clock	10	5	–	ns
t_{su6}	–	28	Setup Time before Internal Synchronous Clock	0	-5	–	ns
t_{h5}	–	29	Hold Time after External Synchronous Clock	12	6	–	ns
t_{h6}	–	30	Hold Time after Internal Synchronous Clock	20	15	–	ns
t_{wh3}, t_{wl3}	–	31,32	Clock Pulse Duration, High, Low	10	8	–	ns

1. External Parameters are tested and guaranteed.
2. See Timing Technical Note for further details.
3. Unless noted, all parameters use ORP, GRP fanout of four, PTSA, and are measured with 16 outputs switching.
4. Refer to Switching Test Conditions section.

Architectural Description

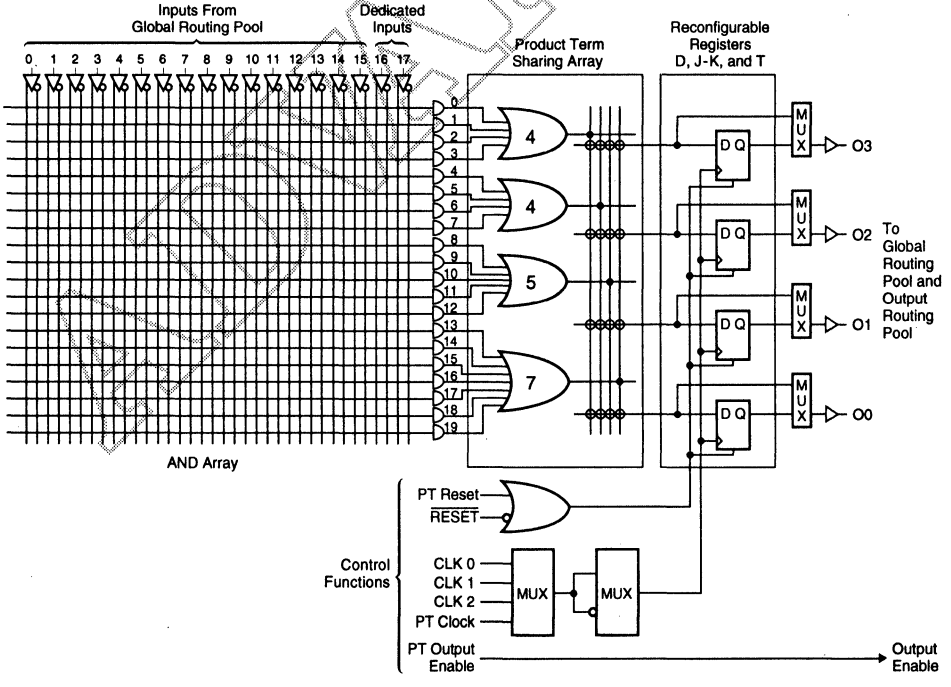
The Generic Logic Block

The Generic Logic Block (GLB) is the standard logic block of the Lattice High Density pLSI Device. This GLB has 18 inputs, four outputs and the logic necessary to implement most standard logic functions. The internal logic of the GLB is divided into four separate sections (see figure 3). The AND Array, the Product Term Sharing Array (PTSA), the Reconfigurable Registers, and the Control Functions. The AND array consists of 20 product terms which can produce the logical sum of any of the 18 GLB inputs. Sixteen of the inputs come from the Global Routing Pool, and are either feedback signals from any of the 16 GLBs or inputs from the external I/O Cells. The two remaining inputs come directly from two dedicated input pins. These signals are available to the product terms in both the logical true and the complemented forms which makes boolean logic reduction easier.

The PTSA takes the 20 product terms and allocates them to the four GLB outputs. There are four OR gates, with four, four, five and seven product terms (see figure 3). The output of any of these gates can be routed to any of the four GLB outputs, and if more product terms are needed, the PTSA can combine them as necessary. If the users' main concern is speed, the PTSA can use a bypass circuit with four product terms to increase the performance of the cell (see figure 4). This can be done to any or all of the four outputs from the GLB.

The Reconfigurable Registers consist of four D-type flip-flops with an Exclusive OR Gate on the input. The Exclusive OR gate in the GLB can be used either as a logic element or to reconfigure the D-type flip-flop to emulate a J-K, or T-type flip-flop (see figure 5). This greatly simplifies the design of counters, comparators and ALU type functions. The registers can be bypassed if the user needs a combinatorial output. Each register output is brought back into the Global Routing Pool and is also brought to the I/O cells via the Output Routing Pool. Reconfigurable registers are not available when the four product term bypass is used.

Figure 3. GLB: Product Term Sharing Array



Architectural Description

Figure 4. GLB: Four Product Term Bypass

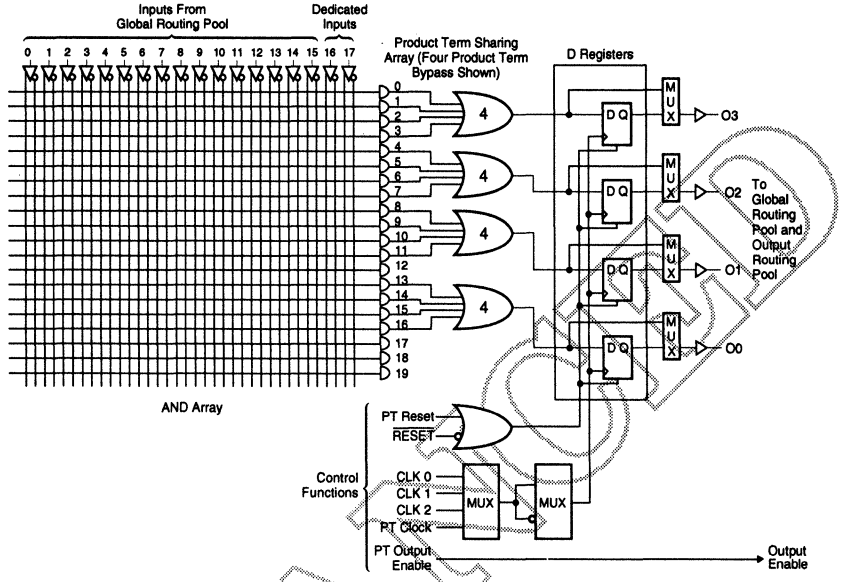
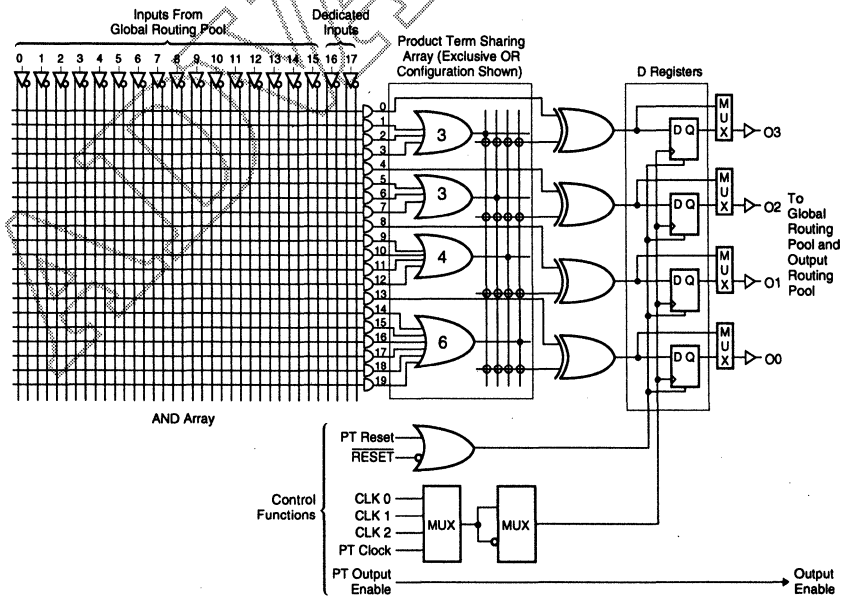


Figure 5. GLB: Exclusive OR Gate



Architectural Description

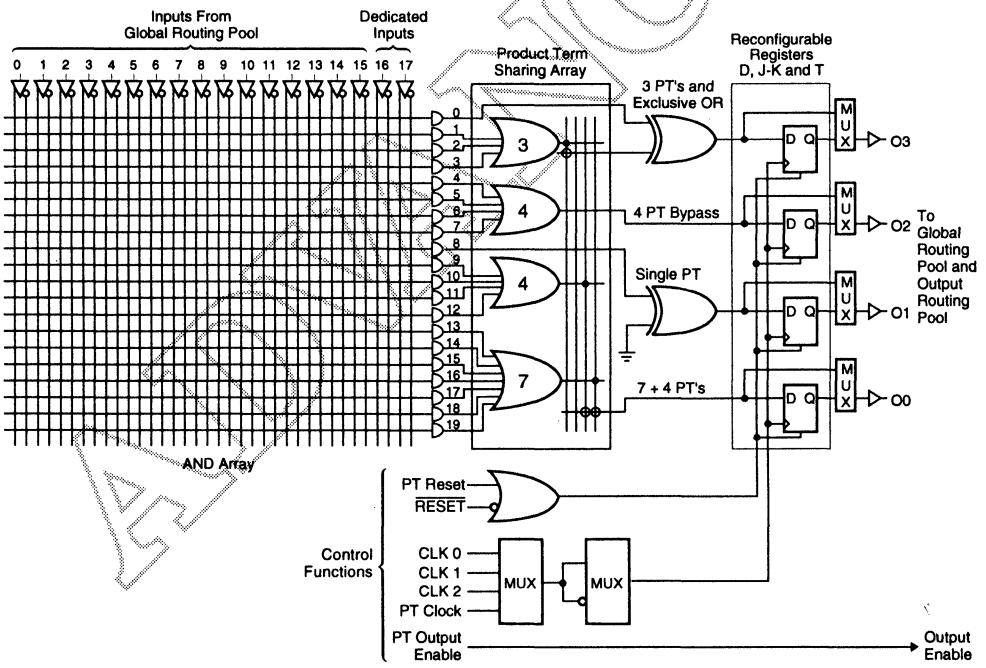
The Generic Logic Block (continued)

The PTSA is flexible enough to allow these features to be used in virtually any combination that the user desires. In the GLB shown in figure 6, Output Three (O3) is configured using the XOR Gate and Output Two (O2) is configured using the four Product Term Bypass. Output One (O1) uses one of the inputs from the five Product Term OR gate while Output Zero (O0) combines the remaining four product terms with all of the product terms from the seven Product Term OR gate for a total of eleven (7+4).

Various signals which control the operation of the GLB are developed in the Control Function section. The clock for the registers can come from any of three sources developed in the Clock Distribution Network (See Clock Distribution Network Section) or from a product term within the GLB. The Reset Signal for the GLB can come from the Global Reset pin or from a product term within the block. The Output Enable for the I/O cells associated with the GLB comes from a product term within the block. Use of a product term for a control function makes that product term unavailable for use as a logic term. Refer to the following table to determine which logic functions are affected.

There are many additional features in a GLB which allow implementation of logic intensive functions. These features are accessible using the hard Macros from the software and require no intervention on the part of the user.

Figure 6. GLB: Various Logical Combinations



Architectural Description

Product Term Sharing Matrix

Product Term #	Standard Configuration Output Number 3 2 1 0	Four Product Term Bypass Output Number 3 2 1 0	Single Product Term Output Number 3 2 1 0	XOR Function Output Number 3 3 2 2 1 1 0 0	Alternate Function
0	■ ■ ■ ■	■	■	■	
1	■ ■ ■ ■	■		■	
2	■ ■ ■ ■	■		■	
3	■ ■ ■ ■	■		■	
4	■ ■ ■ ■	■	■	■	
5	■ ■ ■ ■	■		■	
6	■ ■ ■ ■	■		■	
7	■ ■ ■ ■	■		■	
8	■ ■ ■ ■	■	■	■	
9	■ ■ ■ ■	■		■	
10	■ ■ ■ ■	■		■	
11	■ ■ ■ ■	■		■	
12	■ ■ ■ ■	■		■	■ CLK/Reset
13	■ ■ ■ ■	■	■	■	
14	■ ■ ■ ■	■		■	
15	■ ■ ■ ■	■		■	
16	■ ■ ■ ■	■		■	
17	■ ■ ■ ■			■	
18	■ ■ ■ ■			■	
19	■ ■ ■ ■			■	■ OE/Reset

This matrix shows how each of the product terms are used in the various modes. As an example, Product Term 12 can be used as an input to the five input OR gate in the standard configuration. This OR gate under standard configuration can be routed to any of the four GLB outputs. Product Term 12 is not used in the four product term bypass mode. When GLB output one is used in the Exclusive OR (XOR) mode Product Term 12 becomes one of the inputs to the four input OR Gate. If Product Term 12 is not used in the logic, then it is available for use as either the Asynchronous Clock signal or the GLB Reset signal.

Architectural Description

The Megablock

The pLSI family is structured into multiple "Megablocks". A Megablock consists of 8 GLBs, an Output Routing Pool (ORP) and 16 I/O Cells. Each of these will be explained in detail in the following sections. These elements are coupled together as shown in figure 7. This logic structure is referred to as the Megablock. The various members of the pLSI family are created by combining from two to six Megablocks on a single device.

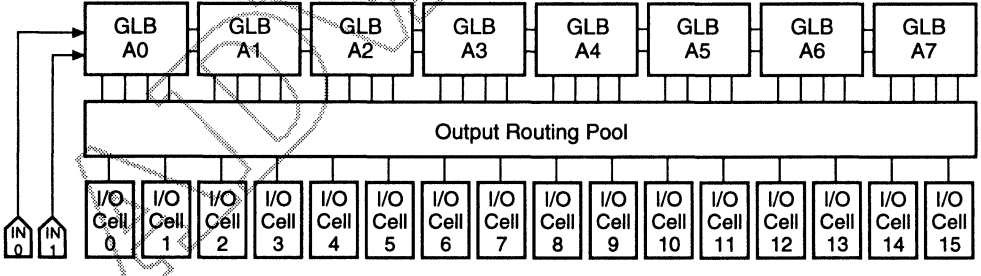
The Megablock shares two sets of resources. The eight GLBs within the Megablock share two dedicated input pins. These dedicated input pins are not available to GLBs in any other Megablock. These pins are dedicated (non-registered) inputs only and are automatically assigned by software. One Output Enable signal is generated within the Megablock and is common to all sixteen of the I/O Cells in the Megablock. The Output Enable signal can be generated using a product term in any of the eight GLBs within the Megablock (see the following section on the Output Enable Multiplexers).

Because of the shared logic within the Megablock, signals which share a common function (counters, busses, etc.) should be grouped within a Megablock. This will allow the user to obtain the best utilization of the logic within the device and eliminate routing bottlenecks.

Megablocks in Each Device

DEVICE	MEGABLOCKS	GLBs	I/O CELLS
pLSI 1016	2	16	32
pLSI 1024	3	24	48
pLSI 1032	4	32	64
pLSI 1048	6	48	96

Figure 7. The Megablock



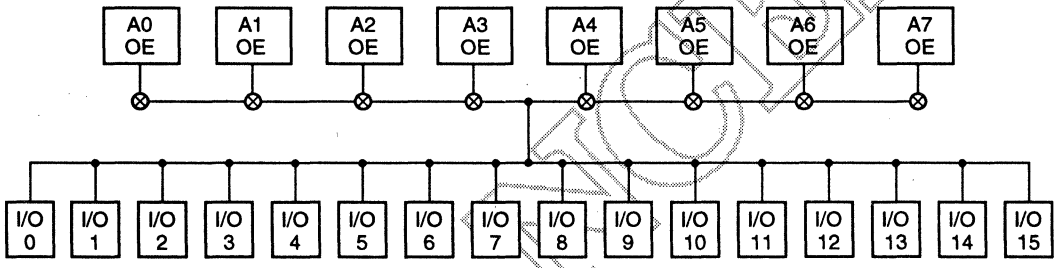
Note: The inputs from the I/O Cell are not shown in this figure, they go directly to the GRP.

Architectural Description

The Output Enable Control

One Output Enable (OE) signal can be generated within each GLB using the OE Product Term 19. One of the eight OE signals from each GLB within a Megablock, is then routed to all of the I/O Cells within the Megablock (see figure 8). This OE signal can simultaneously control all of the 16 I/O cells which are used in 3-state mode. Individual I/O cells also have independent control for permanently enabling or disabling the output buffer (Refer to the I/O Cell Section). Only one OE signal is allowed per Megablock for 3-state operation. The advantage to this approach is that the OE signal can be generated in any GLB within the Megablock which happens to have an unused OE product term. This frees up the other OE product terms for use as logic.

Figure 8. Output Enable Controls



Architectural Description

The Output Routing Pool

The Output Routing Pool (ORP) routes signals from the Generic Logic Block outputs to I/O Cells configured as outputs or bi-directional pins (see figure 9). The purpose of the ORP is to allow greater flexibility when assigning I/O pins. It also simplifies the job for the routing software which results in a higher degree of utilization.

By examining the programmable switch matrix in figure 9, it can be seen that a GLB output can be connected to one of four I/O Cells. Further flexibility is provided by using the PTSA, (Figures 3 through 7) which makes the GLB outputs completely interchangeable. This allows the routing program to freely interchange the outputs to achieve the best routability. This is an automatic process and requires no intervention on the part of the user.

The ORP bypass connections (see figure 10) further increase the flexibility of the device. The ORP bypass connect specific GLB outputs to specific I/O Cells at a faster speed. The bypass path tends to restrict the routability of the device and should only be used for critical signals.

Figure 9. Output Routing Pool

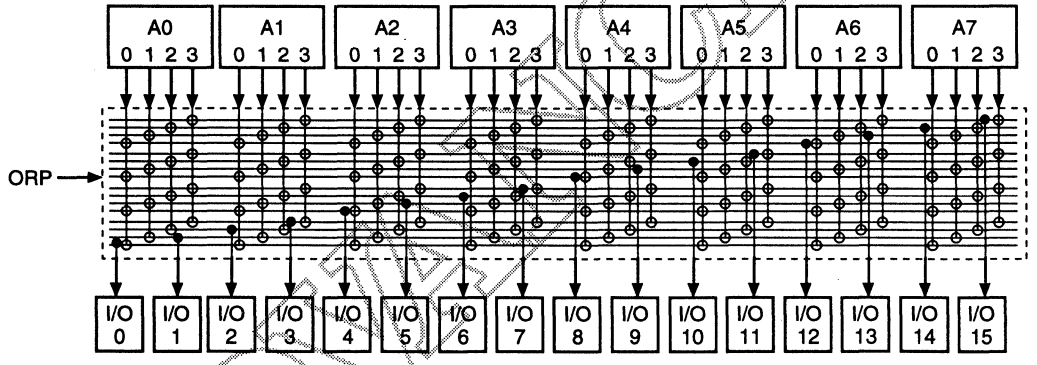
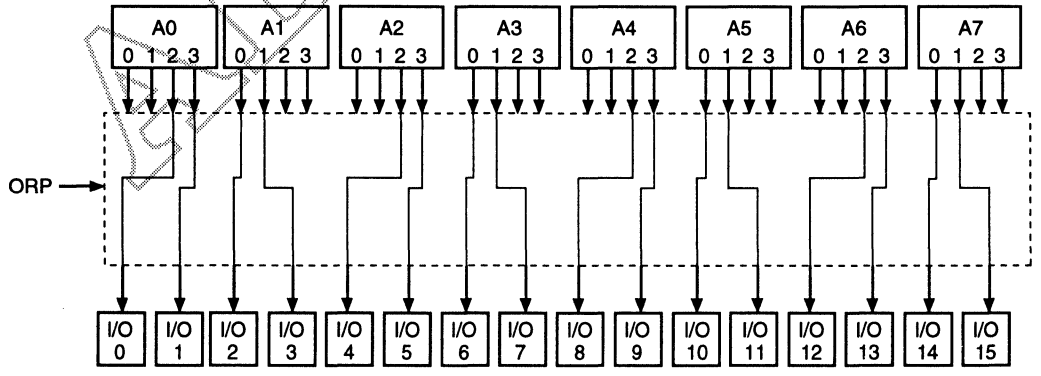


Figure 10. Output Routing Pool showing Bypass



Architectural Description

I/O Cell

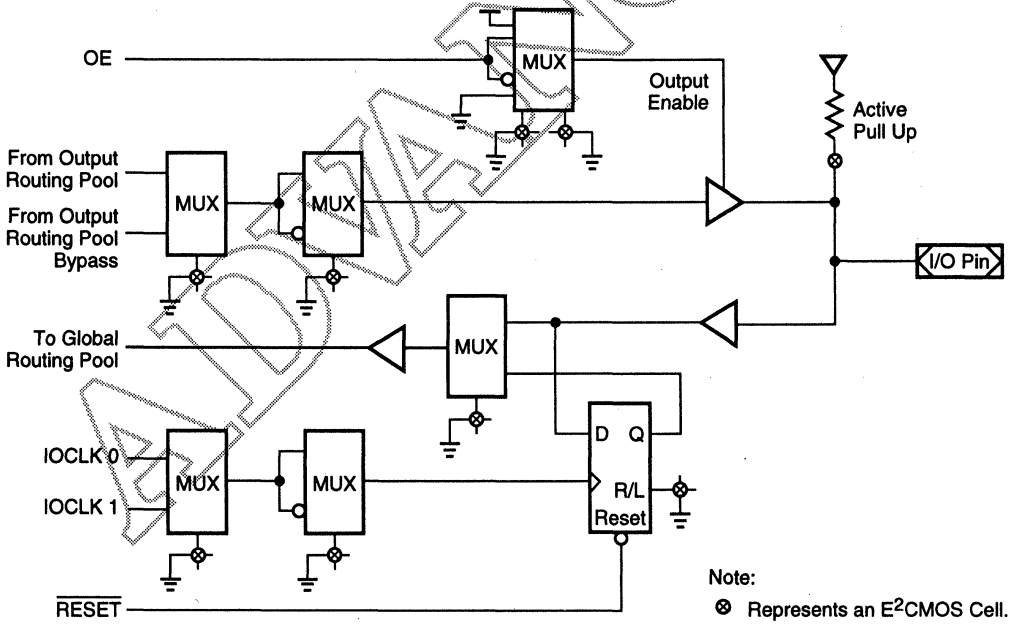
The I/O Cell (see figure 11) is used to route input, output or bidirectional signals connected to the I/O pin. The two logic inputs come from the ORP (see figure 7). One comes from the ORP, and the other comes from the faster ORP bypass. A pair of multiplexers select which signal will be used, and its polarity.

The OE signal comes from the Output Enable. As with the data path, a multiplexer selects the signal polarity. The Output Enable can be set to a logic high (Enabled) when a straight output pin is desired, or logic low (Disabled) when a straight input pin is needed. The Global Reset ($\overline{\text{RESET}}$) signal is driven by the active low chip reset pin. Each I/O Cell can individually select one of the two clock signals (IOCLK 0 or IOCLK 1). These clock signals are generated by the Clock Distribution Network.

Using the multiplexers, the I/O Cell can be configured as an input, an output, a 3-stated output or a bidirectional I/O. The D-type register can be configured as a level sensitive transparent latch or an edge triggered flip-flop to store the in-coming data. Figure 12 illustrates some of the various I/O cell configurations possible.

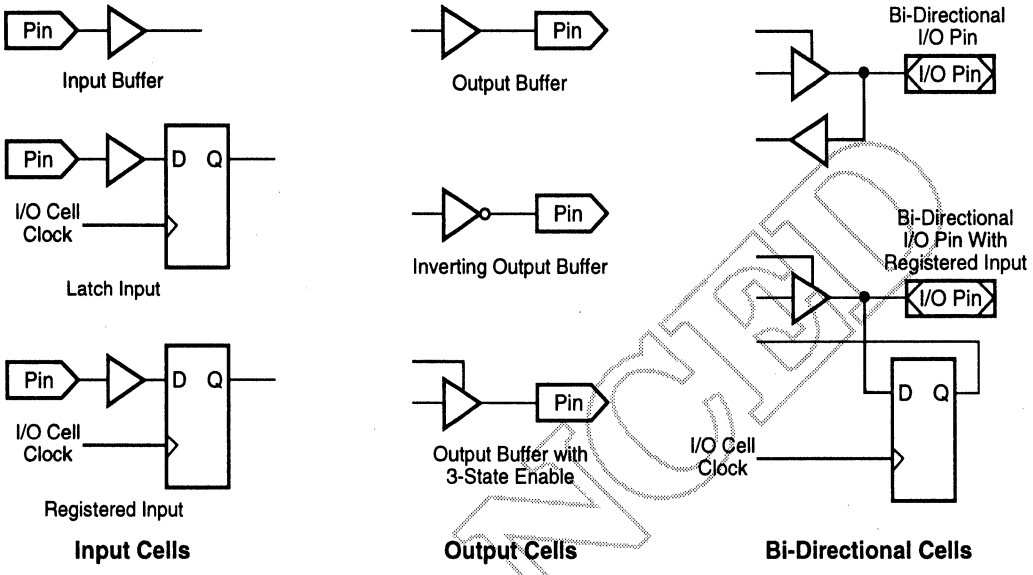
There is an Active Pullup resistor on the I/O pins which is automatically used when the pin is not connected. This prevents the pin from floating and inducing noise into the device or consuming additional power.

Figure 11. I/O Cell Architecture



Architectural Description

Figure 12. Example I/O Cell Configurations



Architectural Description

Clock Distribution Network

The Clock Distribution Network is shown in figure 13. It generates five global clock signals CLK 0, CLK 1, CLK 2 and IOCLK 0, IOCLK 1. The first three, CLK 0, CLK 1 and CLK 2 are used for clocking all the GLBs in the device. Similarly, IOCLK 0 and IOCLK 1 signals are used for clocking all of the I/O Cells in the device. There are three dedicated system clock pins (Y0, Y1, Y2) which can be directed to any GLB or any I/O Cell using the Clock Distribution Network. The other inputs to the Clock Distribution Network are the 4 outputs of a dedicated clock GLB ("B0" for pLSI 1016). These Clock GLB outputs can be used to create a user-defined internal clocking scheme.

Typically the clock GLB will be clocked using an external main clock pin Y0 connected to global clock signal CLK 0. The outputs of the clock GLB in turn will generate "divide by two" or "divide by four" phases of the CLK 0 which can be connected to CLK 1, CLK 2 or IOCLK 0, IOCLK 1 global clocks.

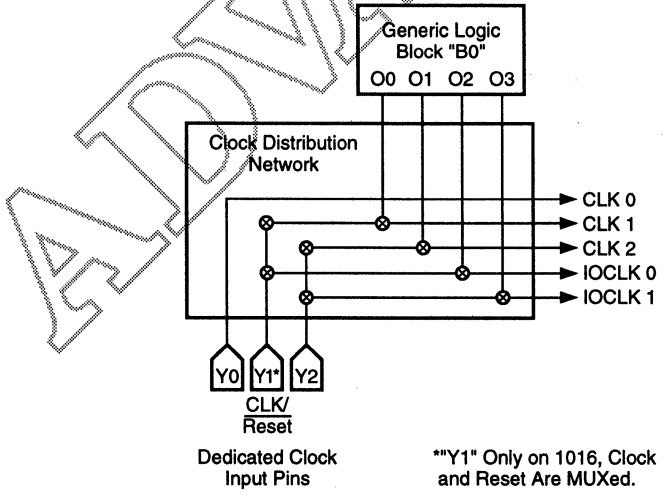
All GLBs also have the capability of generating their own asynchronous clocks using Product Term 12. CLK 0, CLK 1 and CLK 2 feed to their corresponding inputs on all the GLBs (see figure 3).

The two I/O clocks generated in the Clock Distribution Network IOCLK 0 and IOCLK 1, are brought to all 32 of the I/O cells and the user programs the I/O cell to use one of the two.

Global Routing Pool

The Global Routing Pool (GRP) is a Lattice proprietary interconnect structure which offers fast predictable speeds with complete connectivity. The GRP allows the outputs from the GLBs or the I/O cell inputs to be connected to the inputs of the GLBs. Any GLB output is available to the input of all other GLBs, and similarly an input from an I/O pin is available as an input to all of the GLBs. Because of the uniform architecture of the pLSI device, the delays through the Global Routing Pool are both consistent and predictable. However, they are slightly affected by fanout.

Figure 13. Clock Distribution Network



Security Cell

A security cell is provided in the pLSI devices to prevent unauthorized copying of the array patterns. Once programmed, this cell prevents further read access to the functional bits in the device. This cell can only be erased by reprogramming the device, so the original configuration can never be examined once this cell is programmed.

Device Programming

pLSI devices are programmed using a Lattice-approved Device Programmer, available from a number of third party manufacturers. Complete programming of the device takes only a few seconds. Erasing of the device is automatic and is completely transparent to the user. In-system programming is available with ispLSI devices using Lattice programming algorithm.

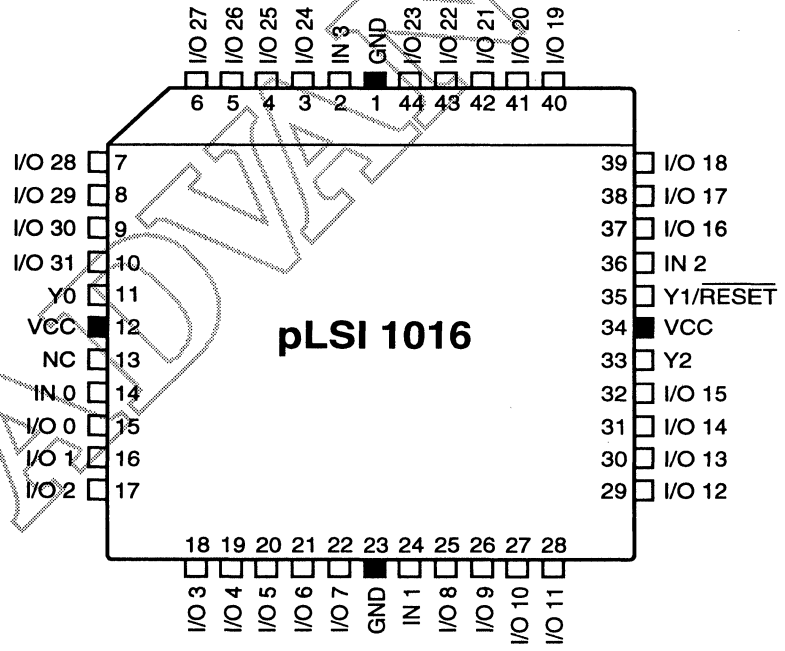
Latch-up Protection

pLSI devices are designed with an on-board charge pump to negatively bias the substrate. The negative bias is of sufficient magnitude to prevent input undershoots from causing the internal circuitry to latchup. Additionally, outputs are designed with n-channel pull-ups instead of the traditional p-channel pull-ups to eliminate any possibility of SCR induced latching.

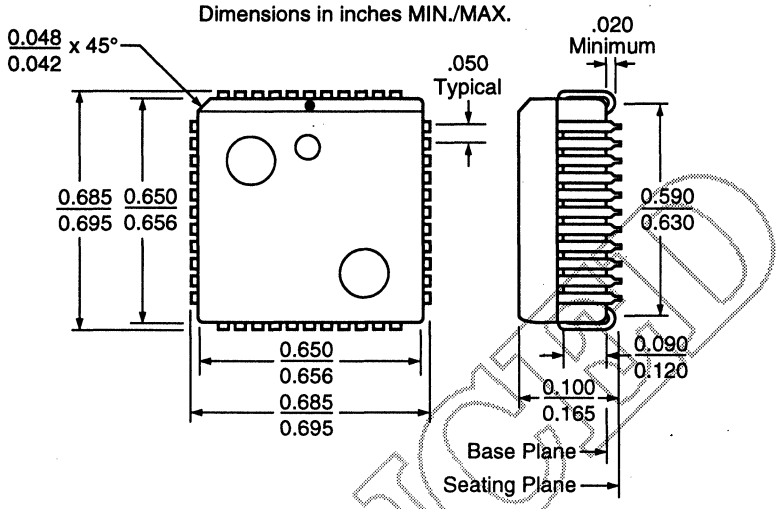
2

Pin Configuration

pLSI 1016 PLCC Pinout Diagram



44-Pin PLCC



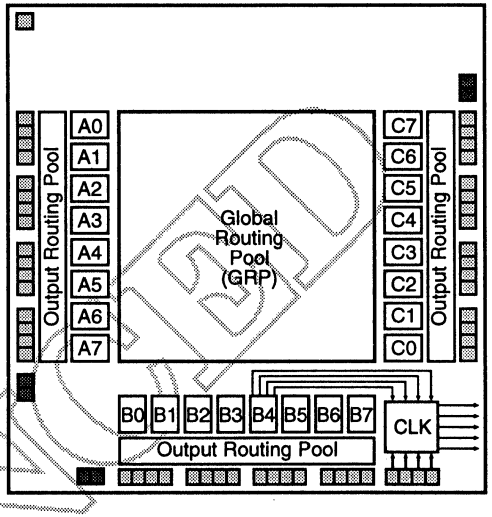
pLSI™ 1024

programmable Large Scale Integration

Features

- **PROGRAMMABLE HIGH DENSITY LOGIC**
  - — Member of Lattice's pLSI Family
  - — High Speed Global Interconnects
  - — 48 I/O Pins, Six Dedicated Inputs
  - — 144 Registers
  - — Wide Input Gating for Fast Counters, State Machines, Address Decoders, etc.
  - — Small Logic Block Size for Random Logic
  - — Security Cell Prevents Unauthorized Copying
- **HIGH PERFORMANCE E²CMOS® TECHNOLOGY**
  - — f_{max} = 80 MHz Maximum Operating Frequency
  - — t_{pd} = 15 ns Propagation Delay
  - — TTL Compatible Inputs and Outputs
  - — Electrically Erasable and Re-Programmable
  - — 100% Tested
- **COMBINES EASE OF USE AND THE FAST SYSTEM SPEED OF PLDs WITH THE DENSITY AND FLEXIBILITY OF FIELD PROGRAMMABLE GATE ARRAYS**
  - — Complete Programmable Device can Combine Glue Logic and Structured Designs
  - — 100% Routable at 80% Utilization
  - — Four Dedicated Clock Input Pins
  - — Synchronous and Asynchronous Clocks
  - — Flexible Pin Placement
  - — Optimized Global Routing Pool Allows Global Interconnectivity
- **pLSI/ispLSI™ DEVELOPMENT SYSTEM (pDS™)**
  - — Boolean Logic Compiler
  - — Automatic Place and Route
  - — Manual Partitioning
  - — PC Platform
  - — Easy to Use Windows Interface
- **ADVANCED pLSI/ispLSI DEVELOPMENT SYSTEM**
  - — Industry Standard, Third Party Design Environments
  - — Schematic Capture
  - — Fully Automatic Partitioning
  - — Automatic Place and Route
  - — Comprehensive Logic and Timing Simulation
  - — PC and Workstation Platforms

Functional Block Diagram



Description

The Lattice pLSI 1024 is a High Density Programmable Logic Device which contains 144 Registers, 48 Universal I/O pins, six Dedicated Input Pins, four Dedicated Clock Input Pins and a Global Routing Pool (GRP). The GRP provides complete interconnectivity between all of these elements.

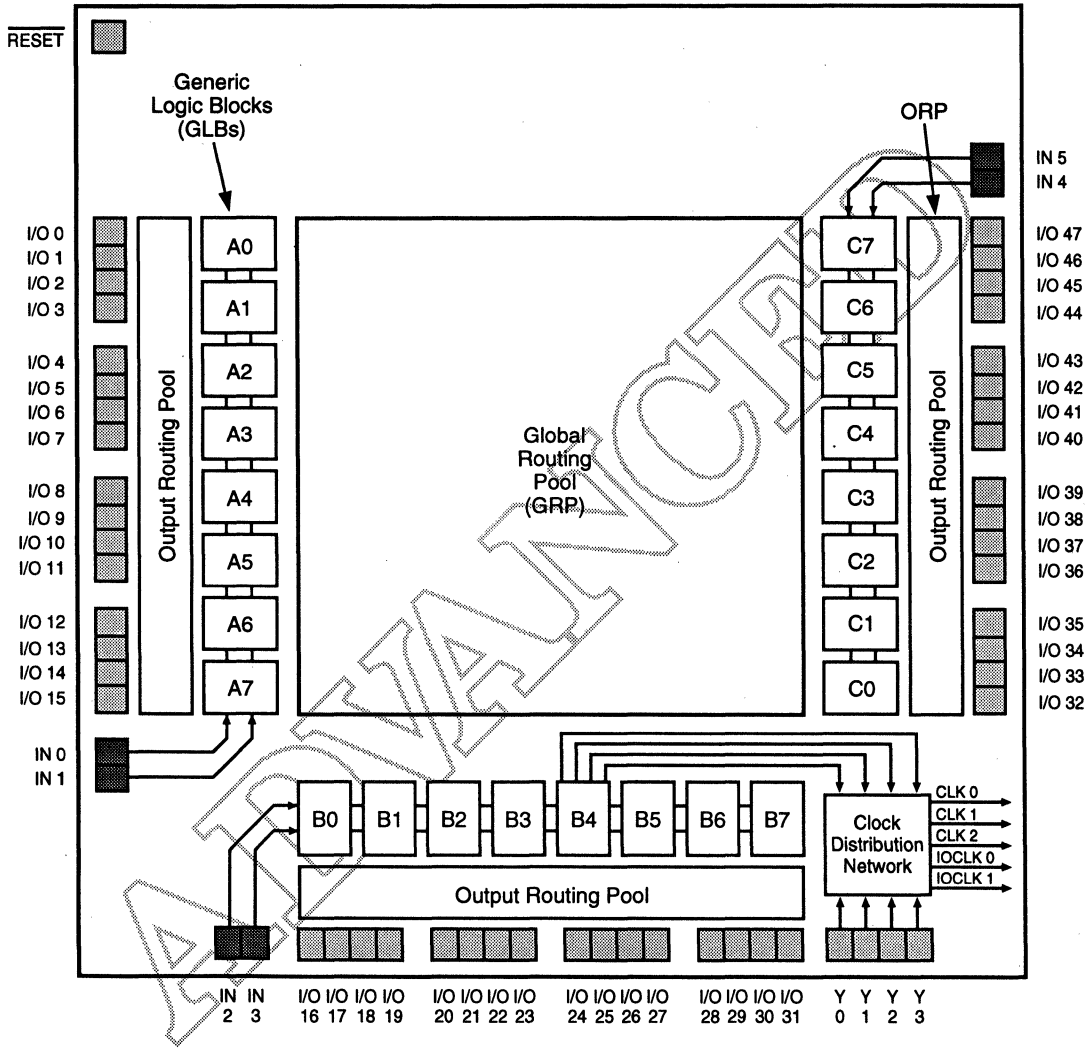
The basic unit of logic on the pLSI 1024 device is the Generic Logic Block (GLB). The GLBs are labeled A0, A1 ..C7, (see figure 1). There are a total of 24 GLBs in the pLSI 1024 device. Each GLB has 18 inputs, a programmable AND/OR/Exclusive OR array, and four outputs which can be configured to be either combinatorial or registered. Inputs to the GLB come from the GRP. All of the GLB outputs are brought back into the GRP so that they can be connected to the inputs of any other GLB on the device.

LATTICE SEMICONDUCTOR CORP., 5555 Northeast Moore Ct., Hillsboro, Oregon 97124, U.S.A.
Tel. 1-800-LATTICE (528-8423); FAX (503) 681-3037

January 1992. Rev. A

Functional Block Diagram

Figure 1. pLSI 1024



Description (continued)

The device also has 48 I/O Cells, each of which is directly connected to an I/O pin. Each I/O cell can be individually programmed to be a combinatorial input, registered input, latched input, output or bi-directional I/O pin with 3-state. The signal levels are TTL compatible voltages and the output drivers can source 4 mA or sink 8 mA.

The 48 I/O Cells are grouped into three sets of 16 each as shown in figure 1. Each of these I/O groups is associated with a logic Megablock through the use of the Output Routing Pool (ORP) and shares a common Output Enable (OE) signal.

Eight GLBs, 16 I/O Cells and one ORP are connected together to make a logic Megablock. The Megablock is defined by the resources that it shares. The outputs of the eight GLBs are connected to a set of 16 universal I/O cells by the ORP. The pLSI 1024 Device contains three of these Megablocks.

The GRP has as its inputs the outputs from all of the GLBs and all of the inputs from the bi-directional I/O cells. All of these signals are made available to the inputs of the GLBs. Delays through the GRP have been equalized to minimize timing skew and logic glitching.

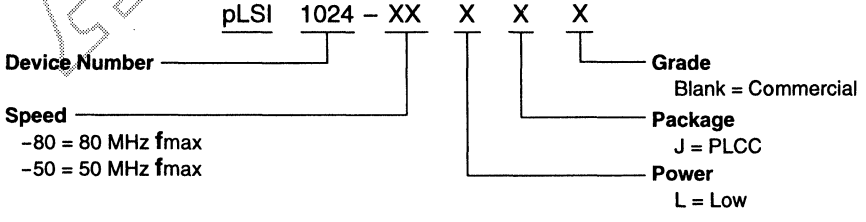
Clocks in the pLSI 1024 device are selected using the Clock Distribution Network. Four dedicated clock pins (Y0 to Y3) are brought into the distribution network, and five outputs (CLK 0 to CLK 2 and IOCLK 0, IOCLK 1) are provided to route clocks to the GLBs and I/O cells. The Clock Distribution Network can also be driven from a special GLB (B4 on the pLSI 1024 device). The logic of this GLB allows the user to create an internal clock from a combination of internal signals within the device.

The pLSI 1024 device is part of Lattice's programmable Large Scale Integration (pLSI) family. This family contains a range of devices from the pLSI 1016, with 96 registers, to the pLSI 1048 with 288 registers. The pLSI Family Product Selector Guide below lists key attributes of the devices along with the number of resources available.

pLSI Family Product Selector Guide

DEVICE	pLSI 1016	pLSI 1024	pLSI 1032	pLSI 1048
GLBs	16	24	32	48
Registers	96	144	192	288
I/O Pins	32	48	64	96
Dedicated Inputs	4	6	8	10
Pin Count	44	68	84	120

Ordering Information



Pin Description

Name	PLCC Pin Numbers	Description
I/O 0 - I/O 3 I/O 4 - I/O 7 I/O 8 - I/O 11 I/O 12 - I/O 15 I/O 16 - I/O 19 I/O 20 - I/O 23 I/O 24 - I/O 27 I/O 28 - I/O 31 I/O 32 - I/O 35 I/O 36 - I/O 39 I/O 40 - I/O 43 I/O 44 - I/O 47	22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14	Input/Output Pins - These are the general purpose I/O pins used by the logic array.
IN 0 - IN 3 IN 4 - IN 5	21, 34, 49, 55, 2, 15	Dedicated input pins to the device.
$\overline{\text{RESET}}$	20	Active Low (0) Reset pin which resets all of the GLB and I/O registers in the device.
Y0	16	Dedicated Clock input. This clock input is connected to one of the clock inputs of all of the GLBs on the device.
Y1	54	Dedicated clock input. This clock input is brought into the clock distribution network, and can optionally be routed to any GLB on the device.
Y2	51	Dedicated clock input. This clock input is brought into the clock distribution network, and can optionally be routed to any GLB and/or any I/O Cell on the device.
Y3	50	Dedicated clock input. This clock input is brought into the clock distribution network, and can optionally be routed to any I/O Cell on the device.
NC	19	This is a factory test pin and it should be left floating or tied to V_{CC}.
GND	1, 18, 35, 52	Ground (GND)
VCC	17, 36, 53, 68	V_{CC}

Absolute Maximum Ratings [1]

Supply Voltage V_{CC} . . . . . -0.5 to +7.0V

Input Voltage Applied. . . . . .-2.5 to V_{CC} +1.0V

Off-State Output Voltage Applied . . . . . .-2.5 to V_{CC} +1.0V

Storage Temperature . . . . . .-65 to 125°C

Ambient Temp. with Power Applied . . . . . .-55 to 125°C

1. Stresses above those listed under the "Absolute Maximum Ratings" may cause permanent damage to the device. Functional operation of the device at these or at any other conditions above those indicated in the operational sections of this specification is not implied (while programming, follow the programming specifications).

DC Recommended Operating Condition

SYMBOL	PARAMETER	MIN.	MAX.	UNITS
T_A	Ambient Temperature	0	70	°C
V_{CC}	Supply Voltage	4.75	5.25	V
V_{IL}	Input Low Voltage	0	0.8	V
V_{IH}	Input High Voltage	2.0	V_{CC}	V

Capacitance (T_A=25°C,f=1.0 MHz)

SYMBOL	PARAMETER	MAXIMUM[1]	UNITS	TEST CONDITIONS
C_1	Input Capacitance	8	pf	V_{CC}=5.0V, V_{IN}=2.0V
C_2	I/O, Y Capacitance	10	pf	V_{CC}=5.0V, VI/O, Y=2.0V

1. Guaranteed but not 100% tested.

Data Retention Specifications

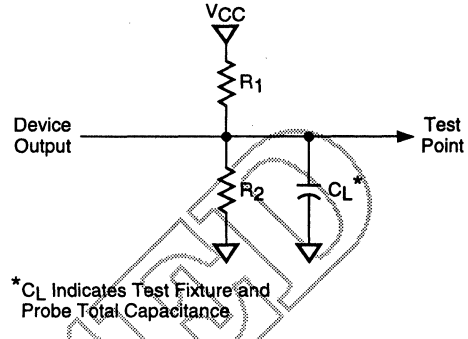
PARAMETER	MINIMUM	MAXIMUM	UNITS
Data Retention	20	–	YEARS
Erase/Reprogram Cycles	–	100	CYCLES

Switching Test Conditions

Input Pulse Levels	GND to 3.0V
Input Rise and Fall Times	3ns 10% to 90%
Input Timing Reference Levels	1.5V
Output Timing Reference Levels	1.5V
Output Load	See Figure 2

3-state levels are measured 0.5V from steady-state active level.

Figure 2. Test Load



Output Load Conditions (see figure 2)

Test Condition		R1	R2	CL
1		470Ω	390Ω	35pF
2	Active High	∞	390Ω	35pF
	Active Low	470Ω	390Ω	35pF
3	Active High to Z at V_{OH} - 0.5V	∞	390Ω	5pF
	Active Low to Z at V_{OL} + 0.5V	470Ω	390Ω	5pF

DC Electrical Characteristics

Over Recommended Operating Conditions

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNITS
V_{OL}	Output Low Voltage	I_{OL} =8 mA.	–	–	0.4	V
V_{OH}	Output High Voltage	I_{OH} =-4 mA.	2.4	–	–	V
I_{IL}	Input or I/O Low Leakage Current	$0V \le V_{IN} \le V_{IL}$ (MAX.)	–	–	-10	µA
I_{IH}	Input or I/O High Leakage Current	$V_{IH} \le V_{IN} \le V_{CC}$	–	–	10	µA
I_{OS}[1]	Output Short Circuit Current	V_{CC} = 5V, V_{OUT} = 0.5V	-60	–	-200	mA
I_{CC}[2]	Operating Power Supply Current	V_{IL} = 0.5V, V_{IH} = 3.0V F_{TOGGLE} = 20 MHz	–	–	–	mA

1. One output at a time for a maximum duration of one second. (Vout = 0.5V)
2. Measured at a frequency of 20 MHz using six 16-bit counters.

External Switching Characteristics[1, 2, 3] pLSI 1024-80

Over Recommended Operating Conditions

PARAMETER	TEST[6] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
t_{pd1}	1	1	Data Propagation Delay, 4PT bypass, ORP bypass	–	12	15	ns
t_{pd2}	1	2	Data Propagation Delay	–	15	20	ns
t_{co1}[5]	1	3	External Clock to Output Delay, ORP bypass	–	8	11	ns
t_{co2}[5]	1	4	External Clock to Output Delay	–	9	14	ns
t_{co3}	1	5	Internal Synch. Clock to Output Delay	–	15	20	ns
t_{co4}	1	6	Asynchronous Clock to Output Delay	–	13	20	ns
t_{r1}	1	7	External Pin Reset to Output Delay	–	13	20	ns
t_{r2}	1	8	Asynchronous PT Reset to Output Delay	–	15	22	ns
t_{en}	2	9	Input to Output Enable		13	20	ns
t_{dis}	3	10	Input to Output Disable	–	13	20	ns

External AC Recommended Operating Conditions[1, 2, 3] pLSI 1024-80

Over Recommended Operating Conditions

PARAMETER	TEST[6] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
f_{max}[4]	1	11	Clock Frequency with Internal Feedback	–	100	80	MHz
f_{max} (External)	1	12	Clock Frequency with External Feedback	–	70	50	MHz
t_{su1}	–	13	Setup Time before External Synch. Clock, 4PT bypass	9	6	–	ns
t_{su2}	–	14	Setup Time before External Synch Clock	12	8	–	ns
t_{su3}	–	15	Setup Time before Internal Synch. Clock	9	3	–	ns
t_{su4}	–	16	Setup Time before Asynchronous Clock	9	4	–	ns
t_{h1}	–	17	Hold time after External Synchronous Clock, 4PT bypass	2	-1	–	ns
t_{h2}	–	18	Hold time after External Synchronous Clock	2	-1	–	ns
t_{h3}	–	19	Hold time after Internal Synchronous Clock	8	2	–	ns
t_{h4}	–	20	Hold time after Asynchronous Clock	8	1	–	ns
t_{rw1}	–	21	External Reset Pulse Duration	10	8	–	ns
t_{rw2}	–	22	Asynchronous Reset Pulse Duration	10	8	–	ns
t_{wh1}, t_{wl1}	–	23,24	External Synchronous Clock Pulse Duration, High, Low	6	5	–	ns
t_{wh2}, t_{wl2}	–	25,26	Asynchronous Clock Pulse Duration, High, Low	6	5	–	ns

1. External Parameters are tested and guaranteed.
2. See Timing Technical Note for further details.
3. Unless noted, all parameters use ORP, GRP fanout of four, PTSA, and are measured with 16 outputs switching.
4. Standard 16-bit counter implementation using GRP feedback.
5. Clock to output specifications include a maximum skew of 2 ns.
6. Refer to Switching Test Conditions section.

Switching Characteristics[1, 2, 3] pLSI 1024-80

Using I/O Cell

PARAMETER	TEST[4] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
t_{su5}	–	27	Setup Time before External Synchronous Clock	5	0	–	ns
t_{su6}	–	28	Setup Time before Internal Synchronous Clock	0	-3	–	ns
t_{h5}	–	29	Hold Time after External Synchronous Clock	8	4	–	ns
t_{h6}	–	30	Hold Time after Internal Synchronous Clock	15	11	–	ns
t_{wh3}, t_{wl3}	–	31,32	Clock Pulse Duration, High, Low	6	5	–	ns

1. External Parameters are tested and guaranteed.
2. See Timing Technical Note for further details.
3. Unless noted, all parameters use ORP, GRP fanout of four, PTSA, and are measured with 16 outputs switching.
4. Refer to Switching Test Conditions section.

External Switching Characteristics[1, 2, 3] pLSI 1024-50

Over Recommended Operating Conditions

PARAMETER	TEST[6] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
t_{pd1}	1	1	Data Propagation Delay, 4PT bypass, ORP bypass	–	16	20	ns
t_{pd2}	1	2	Data Propagation Delay, ORP	–	19	25	ns
t_{co1}[5]	1	3	External Clock to Output Delay, ORP bypass	–	12	16	ns
t_{co2}[5]	1	4	External Clock to Output Delay	–	15	20	ns
t_{co3}	1	5	Internal Synch. Clock to Output Delay	–	21	28	ns
t_{co4}	1	6	Asynchronous Clock to Output Delay	–	21	28	ns
t_{r1}	–	7	External Pin Reset to Output Delay	–	21	28	ns
t_{r2}	–	8	Asynchronous PT Reset to Output Delay	–	24	30	ns
t_{en}	2	9	Input to Output Enable	–	21	28	ns
t_{dis}	3	10	Input to Output Disable	–	21	28	ns

External AC Recommended Operating Conditions[1, 2, 3] pLSI 1024-50

Over Recommended Operating Conditions

PARAMETER	TEST[6] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
f_{max}[4]	1	11	Clock Frequency with Internal Feedback	–	70	50	MHz
f_{max} (External)	1	12	Clock Frequency with External Feedback	–	45	33	MHz
t_{su1}	–	13	Setup Time before External Synch. Clock, 4PT bypass	14	10	–	ns
t_{su2}	–	14	Setup Time before External Synch Clock	17	13	–	ns
t_{su3}	–	15	Setup Time before Internal Synch. Clock	13	9	–	ns
t_{su4}	–	16	Setup Time before Asynchronous Clock	13	9	–	ns
t_{h1}	–	17	Hold time after External Synchronous Clock, 4PT bypass	7	3	–	ns
t_{h2}	–	18	Hold time after External Synchronous Clock	7	3	–	ns
t_{h3}	–	19	Hold time after Internal Synchronous Clock	11	5	–	ns
t_{h4}	–	20	Hold time after Asynchronous Clock	11	5	–	ns
t_{rw1}	–	21	External Reset Pulse Duration	15	13	–	ns
t_{rw2}	–	22	Asynchronous Reset Pulse Duration	15	13	–	ns
t_{wh1}, t_{wl1}	–	23,24	External Synchronous Clock Pulse Duration, High, Low	10	8	–	ns
t_{wh2}, t_{wl2}	–	25,26	Asynchronous Clock Pulse Duration, High, Low	10	8	–	ns

1. External Parameters are tested and guaranteed.
2. See Timing Technical Note for further details.
3. Unless noted, all parameters use ORP, GRP fanout of four, PTSA, and are measured with 16 outputs switching.
4. Standard 16-bit counter implementation using GRP feedback.
5. Clock to output specifications include a maximum skew of 2 ns.
6. Refer to Switching Test Conditions section.

Switching Characteristics[1, 2, 3] — pLSI 1024-50

Using I/O Cell

PARAMETER	TEST[4] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
t_{su5}	–	27	Setup Time before External Synchronous Clock	10	5	–	ns
t_{su6}	–	28	Setup Time before Internal Synchronous Clock	0	-5	–	ns
t_{h5}	–	29	Hold Time after External Synchronous Clock	12	6	–	ns
t_{h6}	–	30	Hold Time after Internal Synchronous Clock	20	15	–	ns
t_{wh3}, t_{wl3}	–	31,32	Clock Pulse Duration, High, Low	10	8	–	ns

1. External Parameters are tested and guaranteed.
2. See Timing Technical Note for further details.
3. Unless noted, all parameters use ORP, GRP fanout of four, PTSA, and are measured with 16 outputs switching.
4. Refer to Switching Test Conditions section.

Architectural Description

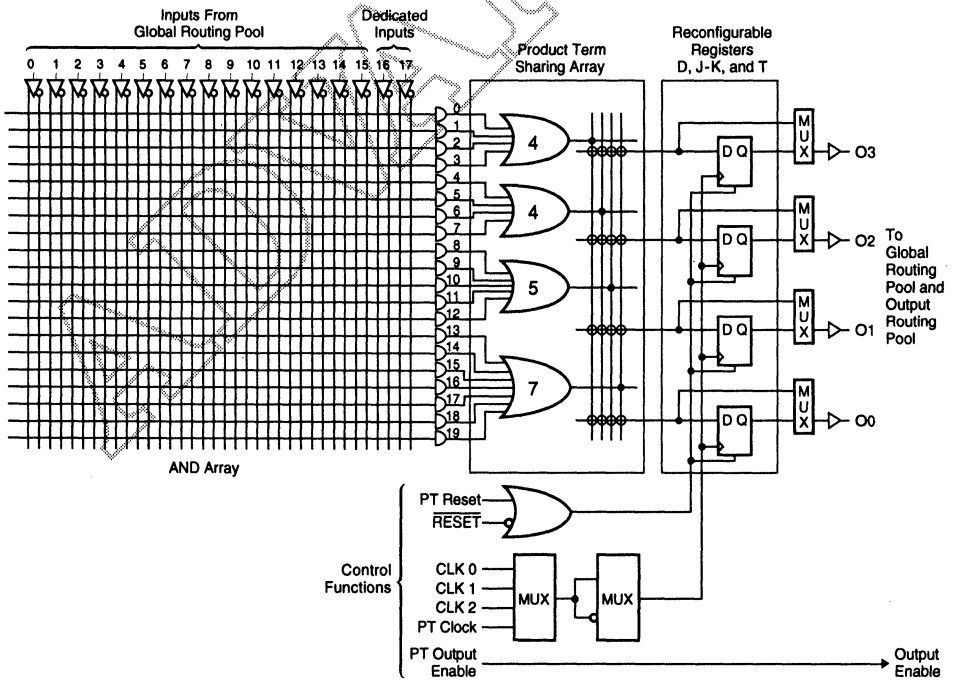
The Generic Logic Block

The Generic Logic Block (GLB) is the standard logic block of the Lattice High Density pLSI Device. This GLB has 18 inputs, four outputs and the logic necessary to implement most standard logic functions. The internal logic of the GLB is divided into four separate sections (see figure 3). The AND Array, the Product Term Sharing Array (PTSA), the Reconfigurable Registers, and the Control Functions. The AND array consists of 20 product terms which can produce the logical sum of any of the 18 GLB inputs. Sixteen of the inputs come from the Global Routing Pool, and are either feedback signals from any of the 24 GLBs or inputs from the external I/O Cells. The two remaining inputs come directly from two dedicated input pins. These signals are available to the product terms in both the logical true and the complemented forms which makes boolean logic reduction easier.

The PTSA takes the 20 product terms and allocates them to the four GLB outputs. There are four OR gates, with four, four, five and seven product terms (see figure 3). The output of any of these gates can be routed to any of the four GLB outputs, and if more product terms are needed, the PTSA can combine them as necessary. If the users' main concern is speed, the PTSA can use a bypass circuit with four product terms to increase the performance of the cell (see figure 4). This can be done to any or all of the four outputs from the GLB.

The Reconfigurable Registers consist of four D-type flip-flops with an Exclusive OR Gate on the input. The Exclusive OR gate in the GLB can be used either as a logic element or to reconfigure the D-type flip-flop to emulate a J-K, or T-type flip-flop (see figure 5). This greatly simplifies the design of counters, comparators and ALU type functions. The registers can be bypassed if the user needs a combinatorial output. Each register output is brought back into the Global Routing Pool and is also brought to the I/O cells via the Output Routing Pool. Reconfigurable registers are not available when the four product term bypass is used.

Figure 3. GLB: Product Term Sharing Array



Architectural Description

Figure 4. GLB: Four Product Term Bypass

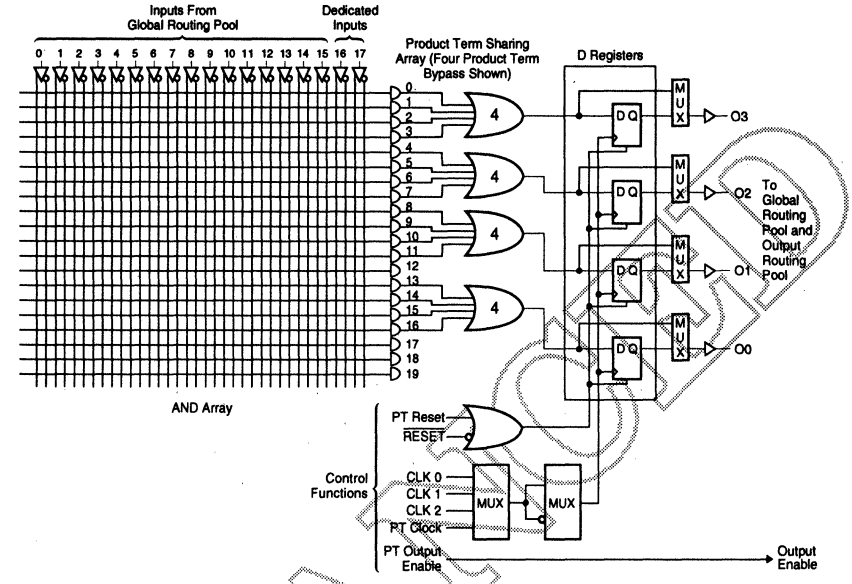
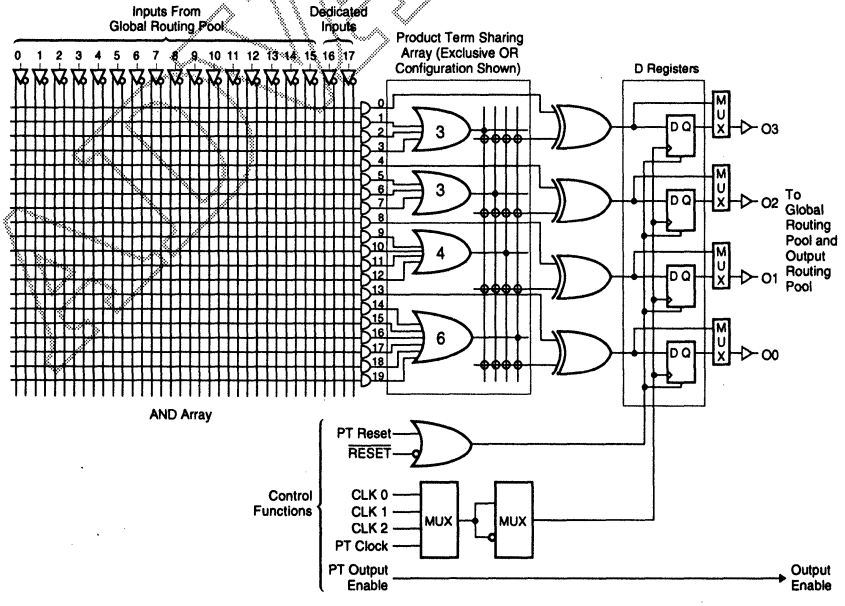


Figure 5. GLB: Exclusive OR Gate



Architectural Description

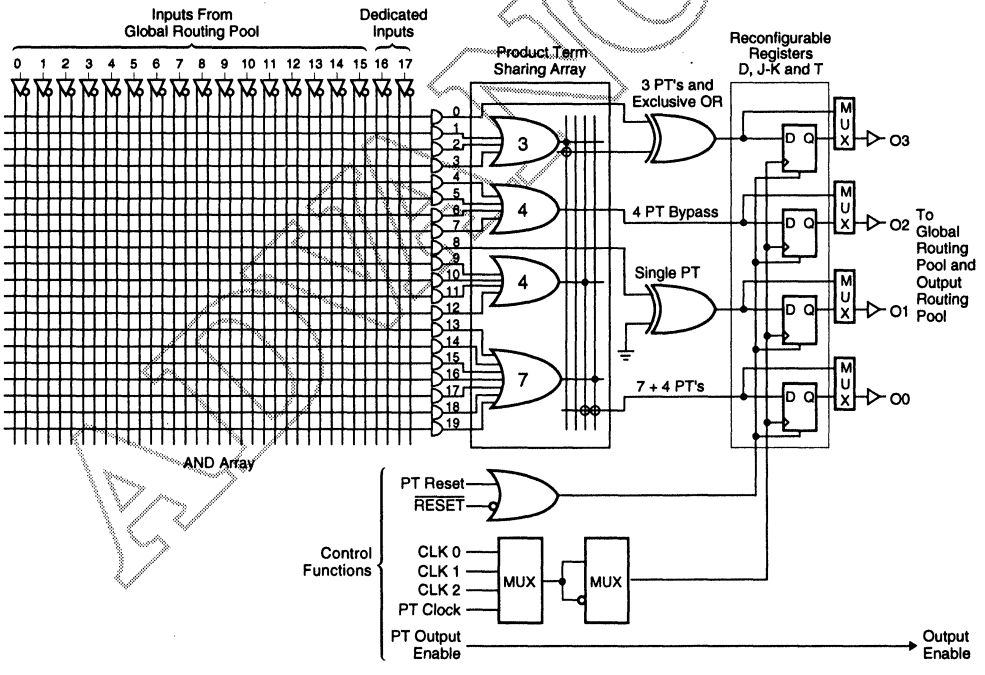
The Generic Logic Block (continued)

The PTSA is flexible enough to allow these features to be used in virtually any combination that the user desires. In the GLB shown in figure 6, Output Three (O3) is configured using the XOR Gate and Output Two (O2) is configured using the four Product Term Bypass. Output One (O1) uses one of the inputs from the five Product Term OR gate while Output Zero (O0) combines the remaining four product terms with all of the product terms from the seven Product Term OR gate for a total of eleven (7+4).

Various signals which control the operation of the GLB are developed in the Control Function section. The clock for the registers can come from any of three sources developed in the Clock Distribution Network (See Clock Distribution Network Section) or from a product term within the GLB. The Reset Signal for the GLB can come from the Global Reset pin or from a product term within the block. The Output Enable for the I/O cells associated with the GLB comes from a product term within the block. Use of a product term for a control function makes that product term unavailable for use as a logic term. Refer to the following table to determine which logic functions are affected.

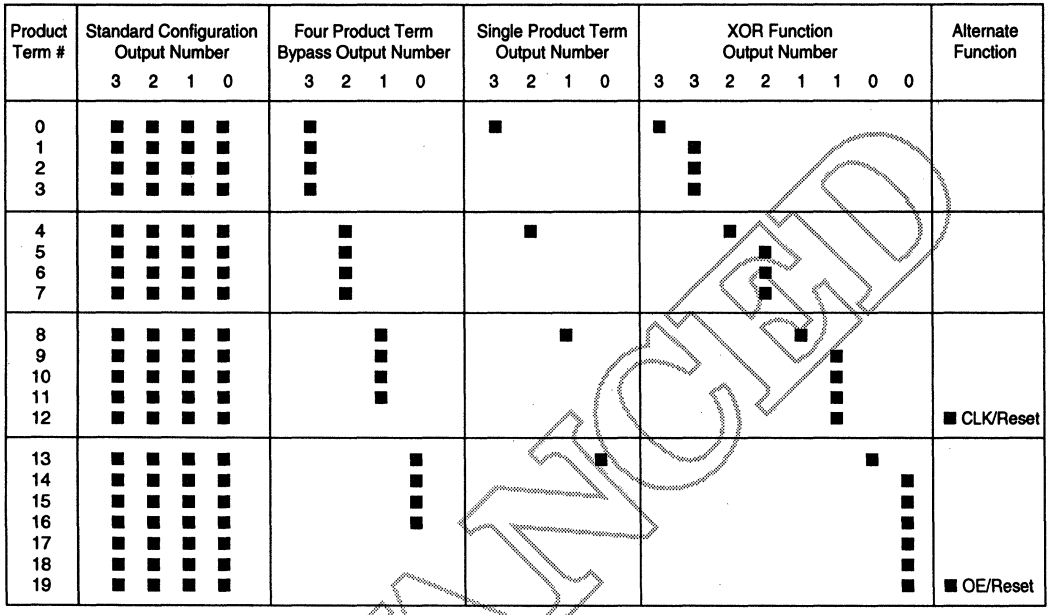
There are many additional features in a GLB which allow implementation of logic intensive functions. These features are accessible using the hard Macros from the software and require no intervention on the part of the user.

Figure 6. GLB: Various Logical Combinations



Architectural Description

Product Term Sharing Matrix



This matrix shows how each of the product terms are used in the various modes. As an example, Product Term 12 can be used as an input to the five input OR gate in the standard configuration. This OR gate under standard configuration can be routed to any of the four GLB outputs. Product Term 12 is not used in the four product term bypass mode. When GLB output one is used in the Exclusive OR (XOR) mode Product Term 12 becomes one of the inputs to the four input OR Gate. If Product Term 12 is not used in the logic, then it is available for use as either the Asynchronous Clock signal or the GLB Reset signal.

Architectural Description

The Megablock

The pLSI family is structured into multiple "Megablocks". A Megablock consists of 8 GLBs, an Output Routing Pool (ORP) and 16 I/O Cells. Each of these will be explained in detail in the following sections. These elements are coupled together as shown in figure 7. This logic structure is referred to as the Megablock. The various members of the pLSI family are created by combining from two to six Megablocks on a single device.

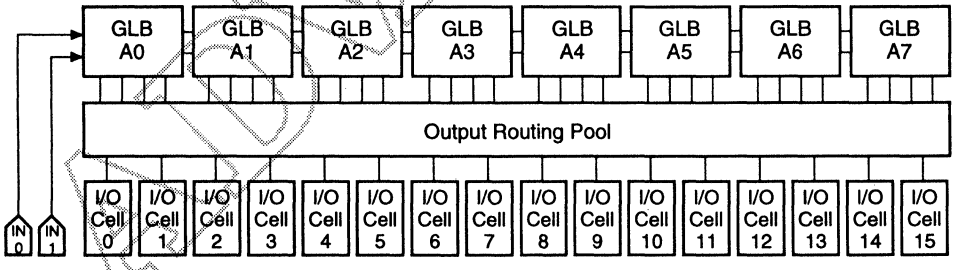
The Megablock shares two sets of resources. The eight GLBs within the Megablock share two dedicated input pins. These dedicated input pins are not available to GLBs in any other Megablock. These pins are dedicated (non-registered) inputs only and are automatically assigned by software. One Output Enable signal is generated within the Megablock and is common to all sixteen of the I/O Cells in the Megablock. The Output Enable signal can be generated using a product term in any of the eight GLBs within the Megablock (see the following section on the Output Enable Multiplexers).

Because of the shared logic within the Megablock, signals which share a common function (counters, busses, etc.) should be grouped within a Megablock. This will allow the user to obtain the best utilization of the logic within the device and eliminate routing bottlenecks.

Megablocks in Each Device

DEVICE	MEGABLOCKS	GLBs	I/O CELLS
pLSI 1016	2	16	32
pLSI 1024	3	24	48
pLSI 1032	4	32	64
pLSI 1048	6	48	96

Figure 7. The Megablock



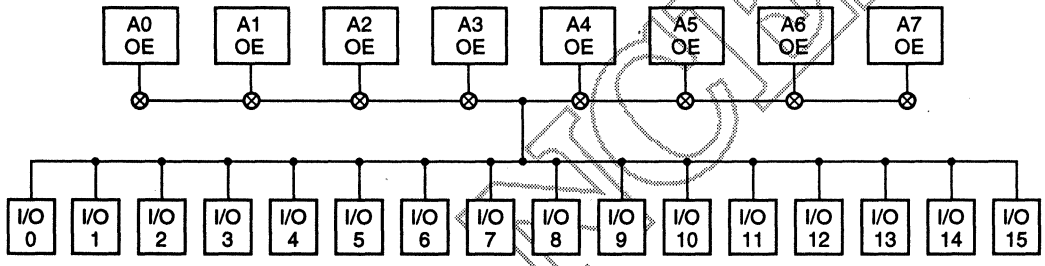
Note: The inputs from the I/O Cell are not shown in this figure, they go directly to the GRP.

Architectural Description

The Output Enable Control

One Output Enable (OE) signal can be generated within each GLB using the OE Product Term 19. One of the eight OE signals from each GLB within a Megablock, is then routed to all of the I/O Cells within the Megablock (see figure 8). This OE signal can simultaneously control all of the 16 I/O cells which are used in 3-state mode. Individual I/O cells also have independent control for permanently enabling or disabling the output buffer (Refer to the I/O Cell Section). Only one OE signal is allowed per Megablock for 3-state operation. The advantage to this approach is that the OE signal can be generated in any GLB within the Megablock which happens to have an unused OE product term. This frees up the other OE product terms for use as logic.

Figure 8. Output Enable Controls



Architectural Description

The Output Routing Pool

The Output Routing Pool (ORP) routes signals from the Generic Logic Block outputs to I/O Cells configured as outputs or bi-directional pins (see figure 9). The purpose of the ORP is to allow greater flexibility when assigning I/O pins. It also simplifies the job for the routing software which results in a higher degree of utilization.

By examining the programmable switch matrix in figure 9, it can be seen that a GLB output can be connected to one of four I/O Cells. Further flexibility is provided by using the PTSA, (Figures 3 through 7) which makes the GLB outputs completely interchangeable. This allows the routing program to freely interchange the outputs to achieve the best routability. This is an automatic process and requires no intervention on the part of the user.

The ORP bypass connections (see figure 10) further increase the flexibility of the device. The ORP bypass connect specific GLB outputs to specific I/O Cells at a faster speed. The bypass path tends to restrict the routability of the device and should only be used for critical signals.

Figure 9. Output Routing Pool

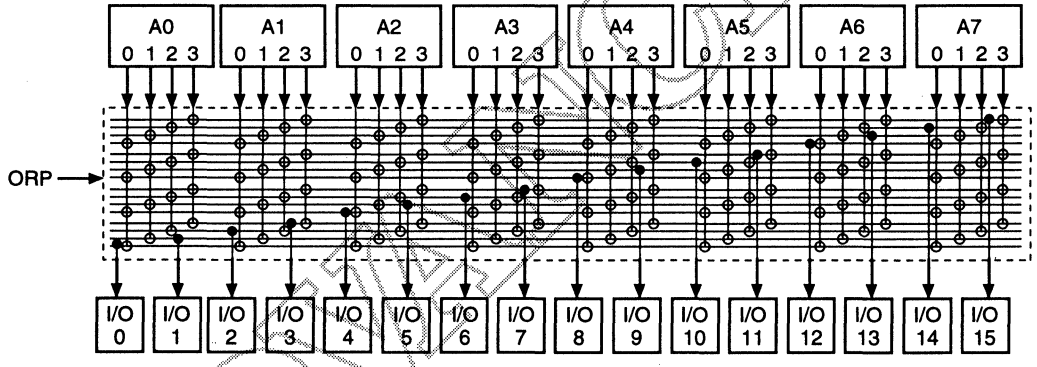
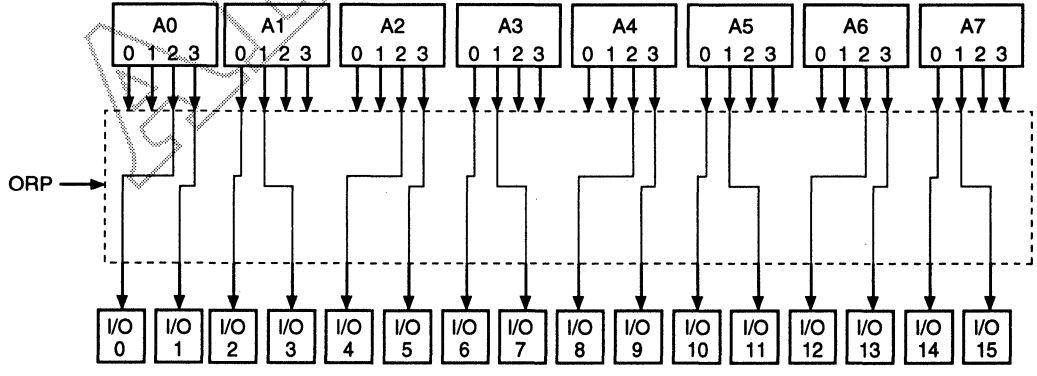


Figure 10. Output Routing Pool showing Bypass



Architectural Description

I/O Cell

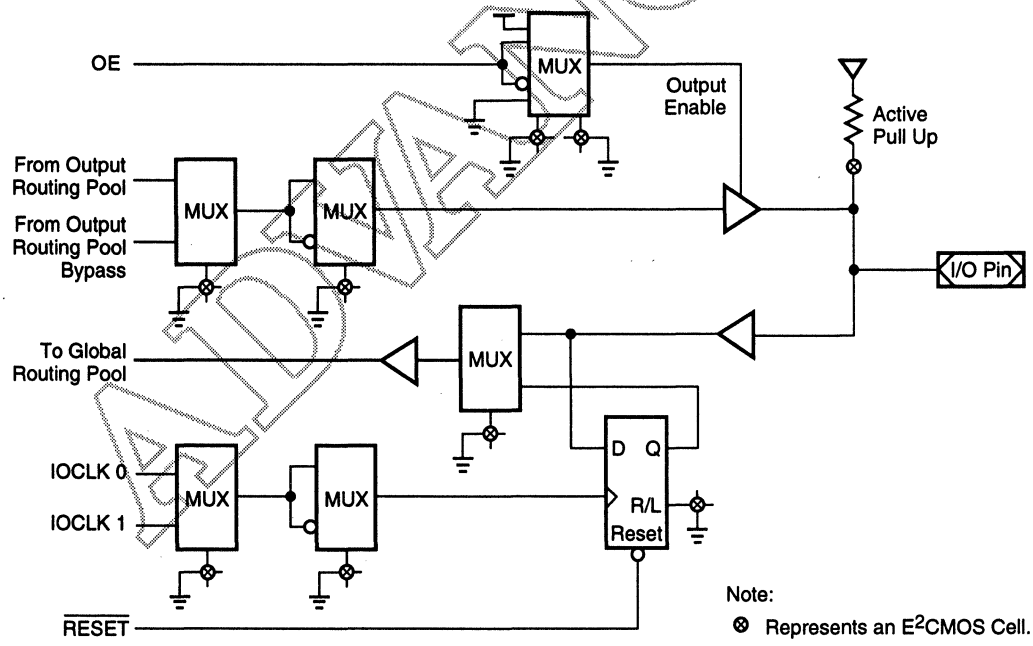
The I/O Cell (see figure 11) is used to route input, output or bidirectional signals connected to the I/O pin. The two logic inputs come from the ORP (see figure 7). One comes from the ORP, and the other comes from the faster ORP bypass. A pair of multiplexers select which signal will be used, and its polarity.

The OE signal comes from the Output Enable. As with the data path, a multiplexer selects the signal polarity. The Output Enable can be set to a logic high (Enabled) when a straight output pin is desired, or logic low (Disabled) when a straight input pin is needed. The Global Reset ($\overline{\text{RESET}}$) signal is driven by the active low chip reset pin. Each I/O Cell can individually select one of the two clock signals (IOCLK 0 or IOCLK 1). These clock signals are generated by the Clock Distribution Network.

Using the multiplexers, the I/O Cell can be configured as an input, an output, a 3-stated output or a bidirectional I/O. The D-type register can be configured as a level sensitive transparent latch or an edge triggered flip-flop to store the in-coming data. Figure 12 illustrates some of the various I/O cell configurations possible.

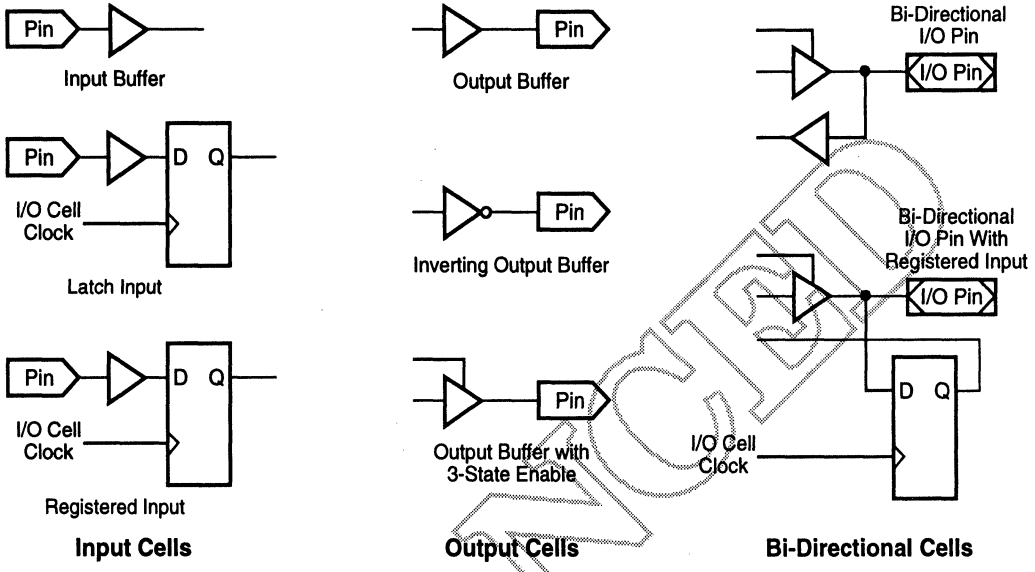
There is an Active Pullup resistor on the I/O pins which is automatically used when the pin is not connected. This prevents the pin from floating and inducing noise into the device or consuming additional power.

Figure 11. I/O Cell Architecture



Architectural Description

Figure 12. Example I/O Cell Configurations



Architectural Description

Clock Distribution Network

The Clock Distribution Network is shown in figure 13. It generates five global clock signals CLK 0, CLK 1, CLK 2 and IOCLK 0, IOCLK 1. The first three, CLK 0, CLK 1 and CLK 2 are used for clocking all the GLBs in the device. Similarly, IOCLK 0 and IOCLK 1 signals are used for clocking all of the I/O Cells in the device. There are four dedicated system clock pins (Y0, Y1, Y2, Y3) which can be directed to any GLB or any I/O Cell using the Clock Distribution Network. The other inputs to the Clock Distribution Network are the 4 outputs of a dedicated clock GLB ("B4" for pLSI 1024). These Clock GLB outputs can be used to create a user-defined internal clocking scheme.

Typically the clock GLB will be clocked using an external main clock pin Y0 connected to global clock signal CLK 0. The outputs of the clock GLB in turn will generate "divide by two" or "divide by four" phases of the CLK 0 which can be connected to CLK 1, CLK 2 or IOCLK 0, IOCLK 1 global clocks.

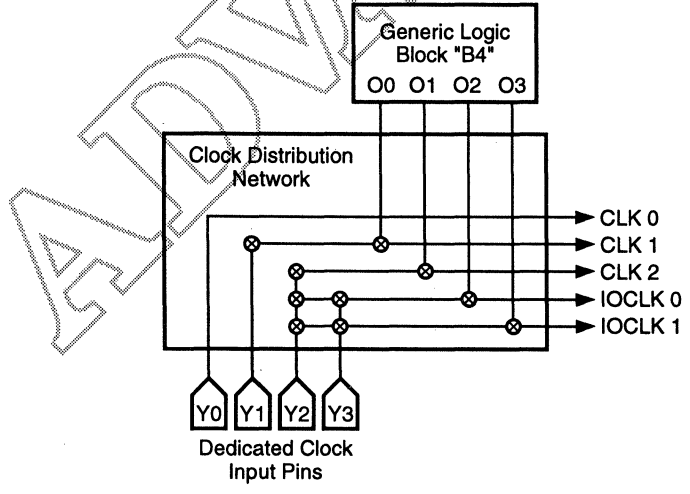
All GLBs also have the capability of generating their own asynchronous clocks using Product Term 12. CLK 0, CLK 1 and CLK 2 feed to their corresponding inputs on all the GLBs (see figure 3).

The two I/O clocks generated in the Clock Distribution Network IOCLK 0 and IOCLK 1, are brought to all 48 of the I/O cells and the user programs the I/O cell to use one of the two.

Global Routing Pool

The Global Routing Pool (GRP) is a Lattice proprietary interconnect structure which offers fast predictable speeds with complete connectivity. The GRP allows the outputs from the GLBs or the I/O cell inputs to be connected to the inputs of the GLBs. Any GLB output is available to the input of all other GLBs, and similarly an input from an I/O pin is available as an input to all of the GLBs. Because of the uniform architecture of the pLSI device, the delays through the Global Routing Pool are both consistent and predictable. However, they are slightly affected by fanout.

Figure 13. Clock Distribution Network



Security Cell

A security cell is provided in the pLSI devices to prevent unauthorized copying of the array patterns. Once programmed, this cell prevents further read access to the functional bits in the device. This cell can only be erased by reprogramming the device, so the original configuration can never be examined once this cell is programmed.

Device Programming

pLSI devices are programmed using a Lattice-approved Device Programmer, available from a number of third party manufacturers. Complete programming of the device takes only a few seconds. Erasing of the device is automatic and is completely transparent to the user. In-system programming is available with ispLSI devices using Lattice programming algorithms.

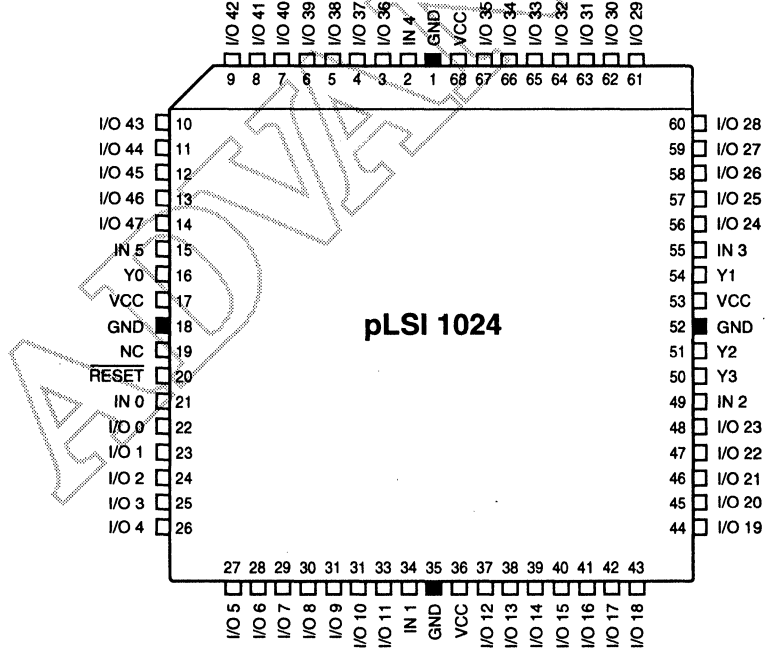
Latch-up Protection

pLSI devices are designed with an on-board charge pump to negatively bias the substrate. The negative bias is of sufficient magnitude to prevent input undershoots from causing the internal circuitry to latchup. Additionally, outputs are designed with n-channel pull-ups instead of the traditional p-channel pull-ups to eliminate any possibility of SCR induced latching.

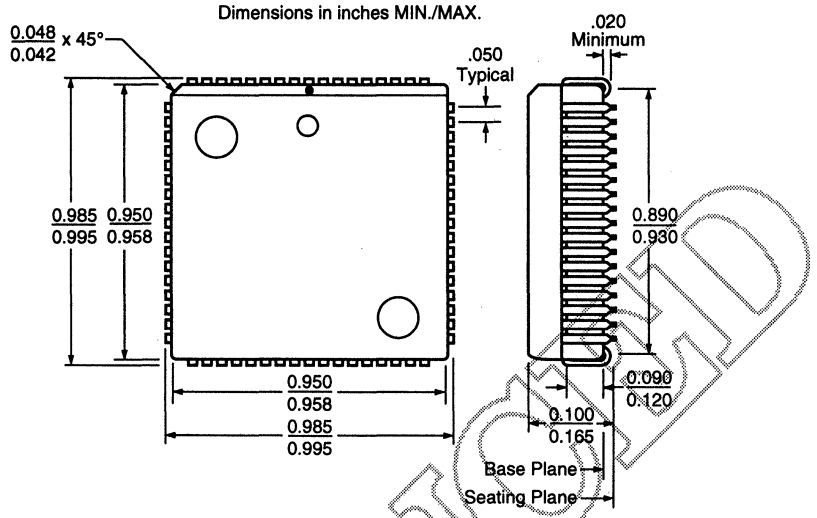
2

Pin Configuration

pLSI 1024 PLCC Pinout Diagram



68-Pin PLCC



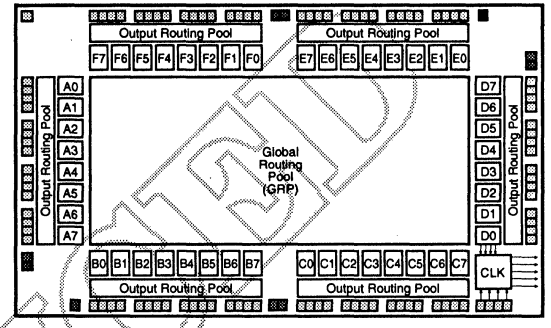
pLSI™ 1048

programmable Large Scale Integration

Features

- **PROGRAMMABLE HIGH DENSITY LOGIC**
  - — Member of Lattice's pLSI Family
  - — High Speed Global Interconnects
  - — 96 I/O Pins, Ten Dedicated Inputs
  - — 288 Registers
  - — Wide Input Gating for Fast Counters, State Machines, Address Decoders, etc.
  - — Small Logic Block Size for Random Logic
  - — Security Cell Prevents Unauthorized Copying
- **HIGH PERFORMANCE E²CMOS® TECHNOLOGY**
  - — f_{max} = 70 MHz Maximum Operating Frequency
  - — t_{pd} = 20 ns Propagation Delay
  - — TTL Compatible Inputs and Outputs
  - — Electrically Erasable and Re-Programmable
  - — 100% Tested
- **COMBINES EASE OF USE AND THE FAST SYSTEM SPEED OF PLDs WITH THE DENSITY AND FLEXIBILITY OF FIELD PROGRAMMABLE GATE ARRAYS**
  - — Complete Programmable Device can Combine Glue Logic and Structured Designs
  - — 100% Routable at 80% Utilization
  - — Four Dedicated Clock Input Pins
  - — Synchronous and Asynchronous Clocks
  - — Flexible Pin Placement
  - — Optimized Global Routing Pool Allows Global Interconnectivity
- **pLSI/ispLSI™ DEVELOPMENT SYSTEM (pDS™)**
  - — Boolean Logic Compiler
  - — Automatic Place and Route
  - — Manual Partitioning
  - — PC Platform
  - — Easy to Use Windows Interface
- **ADVANCED pLSI/ispLSI DEVELOPMENT SYSTEM**
  - — Industry Standard, Third Party Design Environments
  - — Schematic Capture
  - — Fully Automatic Partitioning
  - — Automatic Place and Route
  - — Comprehensive Logic and Timing Simulation
  - — PC and Workstation Platforms

Functional Block Diagram



Description

The Lattice pLSI 1048 is a High Density Programmable Logic Device which contains 288 Registers, 96 Universal I/O pins, ten Dedicated Input Pins, four Dedicated Clock Input Pins and a Global Routing Pool (GRP). The GRP provides complete interconnectivity between all of these elements.

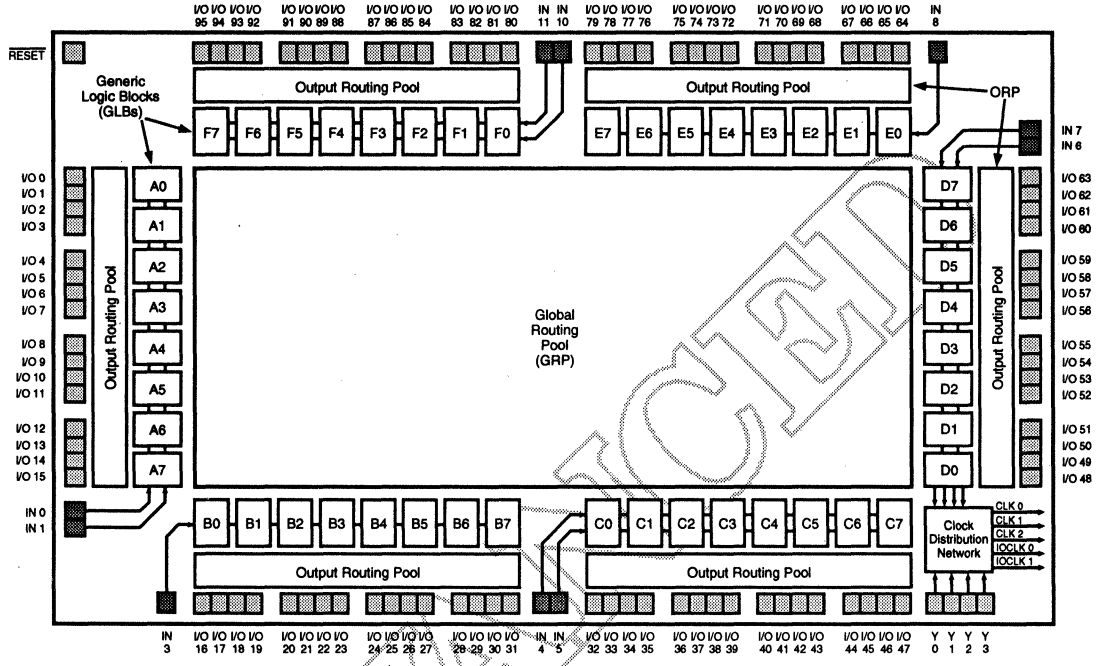
The basic unit of logic on the pLSI 1048 device is the Generic Logic Block (GLB). The GLBs are labeled A0, A1 ..F7, (see figure 1). There are a total of 48 GLBs in the pLSI 1048 device. Each GLB has 18 inputs, a programmable AND/OR/Exclusive OR array, and four outputs which can be configured to be either combinatorial or registered. Inputs to the GLB come from the GRP. All of the GLB outputs are brought back into the GRP so that they can be connected to the inputs of any other GLB on the device.

LATTICE SEMICONDUCTOR CORP., 5555 Northeast Moore Ct., Hillsboro, Oregon 97124, U.S.A.
Tel. 1-800-LATTICE (528-8423); FAX (503) 681-3037

January 1992. Rev. A

Functional Block Diagram

Figure 1. pLSI 1048



Description (continued)

The device also has 96 I/O Cells, each of which is directly connected to an I/O pin. Each I/O cell can be individually programmed to be a combinatorial input, registered input, latched input, output or bi-directional I/O pin with 3-state. The signal levels are TTL compatible voltages and the output drivers can source 4 mA or sink 8 mA.

The 96 I/O Cells are grouped into three sets of 16 each as shown in figure 1. Each of these I/O groups is associated with a logic Megablock through the use of the Output Routing Pool (ORP) and shares a common Output Enable (OE) signal.

Eight GLBs, 16 I/O Cells and one ORP are connected together to make a logic Megablock. The Megablock is defined by the resources that it shares. The outputs of the eight GLBs are connected to a set of 16 universal I/O cells by the ORP. The pLSI 1048 Device contains six of these Megablocks.

The GRP has as its inputs the outputs from all of the GLBs and all of the inputs from the bi-directional I/O cells. All of these signals are made available to the inputs of the GLBs. Delays through the GRP have been equalized to minimize timing skew and logic glitching.

Clocks in the pLSI 1048 device are selected using the Clock Distribution Network. Four dedicated clock pins (Y0 to Y3) are brought into the distribution network, and five outputs (CLK 0 to CLK 2 and IOCLK 0, IOCLK 1) are provided to route clocks to the GLBs and I/O cells. The Clock Distribution Network can also be driven from a special GLB (D0 on the pLSI 1048 device). The logic of this GLB allows the user to create an internal clock from a combination of internal signals within the device.

The pLSI 1048 device is part of Lattice's programmable Large Scale Integration (pLSI) family. This family contains a range of devices from the pLSI 1016, with 96 registers, to the pLSI 1048 with 288 registers. The pLSI Family Product Selector Guide below lists key attributes of the devices along with the number of resources available.

pLSI Family Product Selector Guide

DEVICE	pLSI 1016	pLSI 1024	pLSI 1032	pLSI 1048
GLBs	16	24	32	48
Registers	96	144	192	288
I/O Pins	32	48	64	96
Dedicated Inputs	4	6	8	10
Pin Count	44	68	84	120

Ordering Information

pLSI 1048 – XX X X X

Device Number

Speed

Grade
Blank = Commercial

Package
Q = PQFP

Power
L = Low

Pin Description

Name	PLCC Pin Numbers	Description
I/O 0 - I/O 5 I/O 6 - I/O 11 I/O 12 - I/O 17 I/O 18 - I/O 23 I/O 24 - I/O 29 I/O 30 - I/O 35 I/O 36 - I/O 41 I/O 42 - I/O 47 I/O 48 - I/O 53 I/O 54 - I/O 59 I/O 60 - I/O 65 I/O 66 - I/O 71 I/O 72 - I/O 77 I/O 78 - I/O 83 I/O 84 - I/O 89 I/O 90 - I/O 95	20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99,100,101,102,103, 109,110,111,112,113,114, 115,116,117,118,119,120, 1, 2, 3, 4 5, 6, 7, 8, 9, 10, 11, 12	Input/Output Pins - These are the general purpose I/O pins used by the logic array.
IN 0 - IN 5 IN 6 - IN 11	14, 44, – 47, 48, 73, 79,104,105, – 108, 13	Dedicated input pins to the device. (IN 2 and IN 9 not available)
$\overline{\text{RESET}}$	18	Active Low (0) Reset pin which resets all of the GLB and I/O registers in the device.
Y0	14	Dedicated Clock input. This clock input is connected to one of the clock inputs of all of the GLBs on the device.
Y1	78	Dedicated clock input. This clock input is brought into the clock distribution network, and can optionally be routed to any GLB on the device.
Y2	75	Dedicated clock input. This clock input is brought into the clock distribution network, and can optionally be routed to any GLB and/or any I/O Cell on the device.
Y3	74	Dedicated clock input. This clock input is brought into the clock distribution network, and can optionally be routed to any I/O Cell on the device.
NC	17	This is a factory test pin and it should be loft floating or tied to V_{CC}.
GND	46, 76,106, 16	Ground (GND)
VCC	15, 45, 77, 107	V_{CC}

Absolute Maximum Ratings [1]

Supply Voltage V_{CC} . . . . . -0.5 to +7.0V

Input Voltage Applied. . . . . .-2.5 to V_{CC} +1.0V

Off-State Output Voltage Applied . . . . . .-2.5 to V_{CC} +1.0V

Storage Temperature . . . . . .-65 to 125°C

Ambient Temp. with Power Applied . . . . . .-55 to 125°C

1. Stresses above those listed under the "Absolute Maximum Ratings" may cause permanent damage to the device. Functional operation of the device at these or at any other conditions above those indicated in the operational sections of this specification is not implied (while programming, follow the programming specifications).

DC Recommended Operating Condition

SYMBOL	PARAMETER	MIN.	MAX.	UNITS
T_A	Ambient Temperature	0	70	°C
V_{CC}	Supply Voltage	4.75	5.25	V
V_{IL}	Input Low Voltage	0	0.8	V
V_{IH}	Input High Voltage	2.0	V_{CC}	V

Capacitance (T_A=25°C,f=1.0 MHz)

SYMBOL	PARAMETER	MAXIMUM[1]	UNITS	TEST CONDITIONS
C_1	Input Capacitance	8	pf	V_{CC}=5.0V, V_{IN}=2.0V
C_2	I/O, Y Capacitance	10	pf	V_{CC}=5.0V, VI/O, Y=2.0V

1. Guaranteed but not 100% tested.

Data Retention Specifications

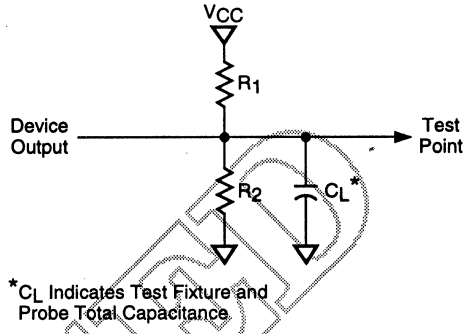
PARAMETER	MINIMUM	MAXIMUM	UNITS
Data Retention	20	–	YEARS
Erase/Reprogram Cycles	–	100	CYCLES

Switching Test Conditions

Input Pulse Levels	GND to 3.0V
Input Rise and Fall Times	3ns 10% to 90%
Input Timing Reference Levels	1.5V
Output Timing Reference Levels	1.5V
Output Load	See Figure 2

3-state levels are measured 0.5V from steady-state active level.

Figure 2. Test Load



Output Load Conditions (see figure 2)

Test Condition		R1	R2	CL
1		470Ω	390Ω	35pF
2	Active High	∞	390Ω	35pF
	Active Low	470Ω	390Ω	35pF
3	Active High to Z at V_{OH} - 0.5V	∞	390Ω	5pF
	Active Low to Z at V_{OL} + 0.5V	470Ω	390Ω	5pF

DC Electrical Characteristics

Over Recommended Operating Conditions

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNITS
V_{OL}	Output Low Voltage	I_{OL} =8 mA.	–	–	0.4	V
V_{OH}	Output High Voltage	I_{OH} =-4 mA.	2.4	–	–	V
I_{IL}	Input or I/O Low Leakage Current	$0V \le V_{IN} \le V_{IL}$ (MAX.)	–	–	-10	µA
I_{IH}	Input or I/O High Leakage Current	$V_{IH} \le V_{IN} \le V_{CC}$	–	–	10	µA
I_{OS}[1]	Output Short Circuit Current	V_{CC} = 5V, V_{OUT} = 0.5V	-60	–	-200	mA
I_{CC}[2]	Operating Power Supply Current	V_{IL} = 0.5V, V_{IH} = 3.0V F_{TOGGLE} = 20 MHz	–	–	–	mA

1. One output at a time for a maximum duration of one second. (Vout = 0.5V)
2. Measured at a frequency of 20 MHz using twelve 16-bit counters.

External Switching Characteristics[1, 2, 3] pLSI 1048-70

Over Recommended Operating Conditions

PARAMETER	TEST[6] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
t_{pd1}	1	1	Data Propagation Delay, 4PT bypass, ORP bypass	–	–	20	ns
t_{pd2}	1	2	Data Propagation Delay	–	–	25	ns
t_{co1}[5]	1	3	External Clock to Output Delay, ORP bypass	–	–	–	ns
t_{co2}[5]	1	4	External Clock to Output Delay	–	–	–	ns
t_{co3}	1	5	Internal Synch. Clock to Output Delay	–	–	–	ns
t_{co4}	1	6	Asynchronous Clock to Output Delay	–	–	–	ns
t_{r1}	1	7	External Pin Reset to Output Delay	–	–	–	ns
t_{r2}	1	8	Asynchronous PT Reset to Output Delay	–	–	–	ns
t_{en}	2	9	Input to Output Enable	–	–	–	ns
t_{dis}	3	10	Input to Output Disable	–	–	–	ns

External AC Recommended Operating Conditions[1, 2, 3] pLSI 1048-70

Over Recommended Operating Conditions

PARAMETER	TEST[6] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
f_{max}[4]	1	11	Clock Frequency with Internal Feedback	–	–	70	MHz
f_{max} (External)	1	12	Clock Frequency with External Feedback	–	–	–	MHz
t_{su1}	–	13	Setup Time before External Synch. Clock, 4PT bypass	–	–	–	ns
t_{su2}	–	14	Setup Time before External Synch Clock	–	–	–	ns
t_{su3}	–	15	Setup Time before Internal Synch. Clock	–	–	–	ns
t_{su4}	–	16	Setup Time before Asynchronous Clock	–	–	–	ns
t_{h1}	–	17	Hold time after External Synchronous Clock, 4PT bypass	–	–	–	ns
t_{h2}	–	18	Hold time after External Synchronous Clock	–	–	–	ns
t_{h3}	–	19	Hold time after Internal Synchronous Clock	–	–	–	ns
t_{h4}	–	20	Hold time after Asynchronous Clock	–	–	–	ns
t_{rw1}	–	21	External Reset Pulse Duration	–	–	–	ns
t_{rw2}	–	22	Asynchronous Reset Pulse Duration	–	–	–	ns
t_{wh1}, t_{wl1}	–	23,24	External Synchronous Clock Pulse Duration, High, Low	–	–	–	ns
t_{wh2}, t_{wl2}	–	25,26	Asynchronous Clock Pulse Duration, High, Low	–	–	–	ns

1. External Switching Characteristics are tested and guaranteed.
2. See Timing Technical Note for further details.
3. Unless noted, all parameters use ORP, GRP fanout of four, PTSA, and are measured with 16 outputs switching.
4. Standard 16-bit counter implementation using GRP feedback.
5. Clock to output specifications include a maximum skew of 2 ns.
6. Refer to Switching Test Conditions section.

Switching Characteristics[1, 2, 3] — pLSI 1048-80

Using I/O Cell

PARAMETER	TEST[4] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
t_{su5}	–	27	Setup Time before External Synchronous Clock	–	–	–	ns
t_{su6}	–	28	Setup Time before Internal Synchronous Clock	–	–	–	ns
t_{h5}	–	29	Hold Time after External Synchronous Clock	–	–	–	ns
t_{h6}	–	30	Hold Time after Internal Synchronous Clock	–	–	–	ns
t_{wh3}, t_{wl3}	–	31,32	Clock Pulse Duration, High, Low	–	–	–	ns

1. External Parameters are tested and guaranteed.
2. See Timing Technical Note for further details.
3. Unless noted, all parameters use ORP, GRP fanout of four, PTSA, and are measured with 16 outputs switching.
4. Refer to Switching Test Conditions section.

Architectural Description

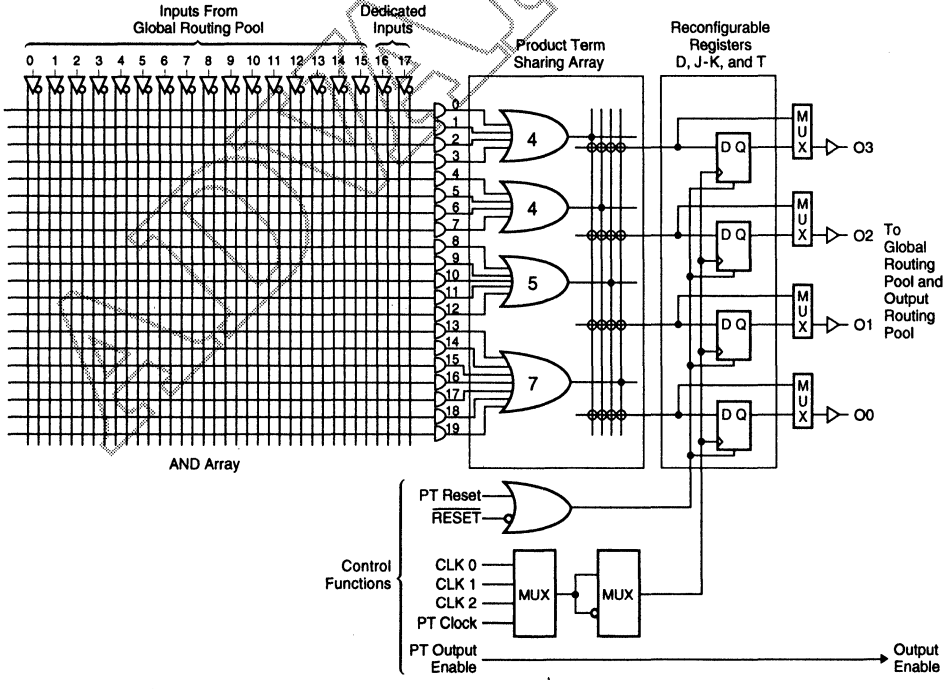
The Generic Logic Block

The Generic Logic Block (GLB) is the standard logic block of the Lattice High Density pLSI Device. This GLB has 18 inputs, four outputs and the logic necessary to implement most standard logic functions. The internal logic of the GLB is divided into four separate sections (see figure 3). The AND Array, the Product Term Sharing Array (PTSA), the Reconfigurable Registers, and the Control Functions. The AND array consists of 20 product terms which can produce the logical sum of any of the 18 GLB inputs. Sixteen of the inputs come from the Global Routing Pool, and are either feedback signals from any of the 48 GLBs or inputs from the external I/O Cells. The two remaining inputs come directly from two dedicated input pins. These signals are available to the product terms in both the logical true and the complemented forms which makes boolean logic reduction easier.

The PTSA takes the 20 product terms and allocates them to the four GLB outputs. There are four OR gates, with four, four, five and seven product terms (see figure 3). The output of any of these gates can be routed to any of the four GLB outputs, and if more product terms are needed, the PTSA can combine them as necessary. If the users' main concern is speed, the PTSA can use a bypass circuit with four product terms to increase the performance of the cell (see figure 4). This can be done to any or all of the four outputs from the GLB.

The Reconfigurable Registers consist of four D-type flip-flops with an Exclusive OR Gate on the input. The Exclusive OR gate in the GLB can be used either as a logic element or to reconfigure the D-type flip-flop to emulate a J-K, or T-type flip-flop (see figure 5). This greatly simplifies the design of counters, comparators and ALU type functions. The registers can be bypassed if the user needs a combinatorial output. Each register output is brought back into the Global Routing Pool and is also brought to the I/O cells via the Output Routing Pool. Reconfigurable registers are not available when the four product term bypass is used.

Figure 3. GLB: Product Term Sharing Array



Architectural Description

Figure 4. GLB: Four Product Term Bypass

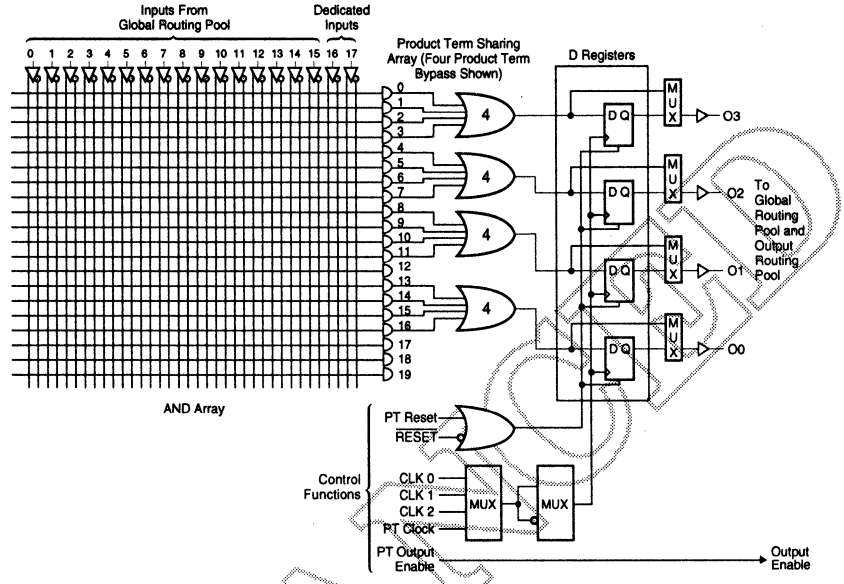
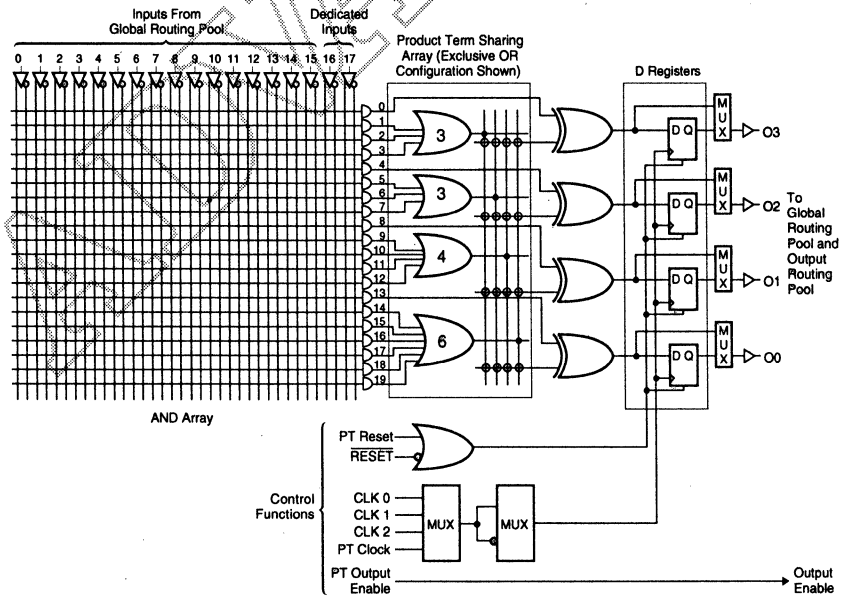


Figure 5. GLB: Exclusive OR Gate



Architectural Description

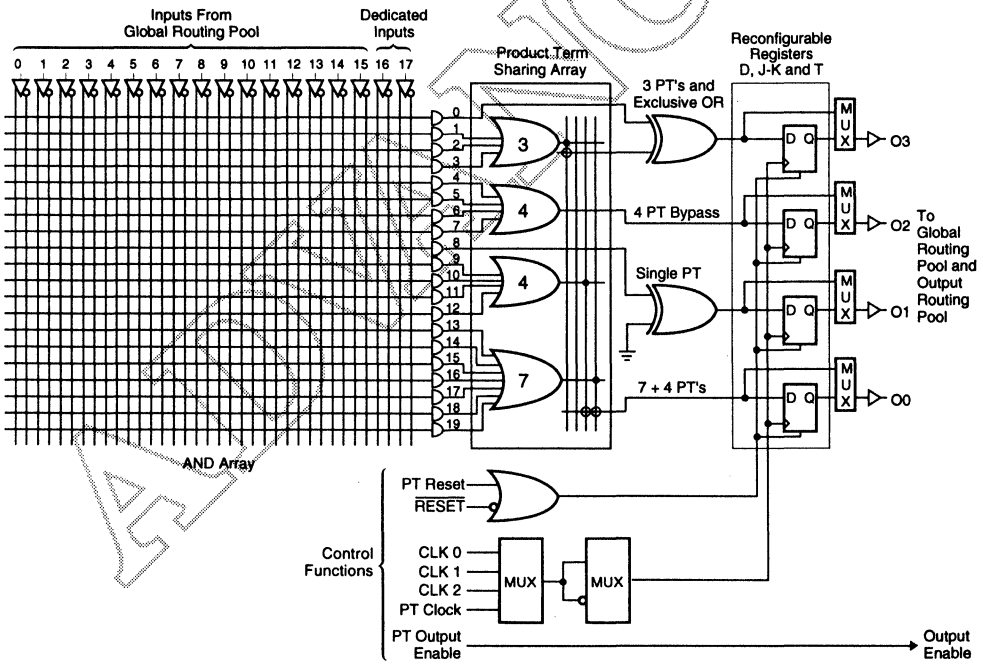
The Generic Logic Block (continued)

The PTSA is flexible enough to allow these features to be used in virtually any combination that the user desires. In the GLB shown in figure 6, Output Three (O3) is configured using the XOR Gate and Output Two (O2) is configured using the four Product Term Bypass. Output One (O1) uses one of the inputs from the five Product Term OR gate while Output Zero (O0) combines the remaining four product terms with all of the product terms from the seven Product Term OR gate for a total of eleven (7+4).

Various signals which control the operation of the GLB are developed in the Control Function section. The clock for the registers can come from any of three sources developed in the Clock Distribution Network (See Clock Distribution Network Section) or from a product term within the GLB. The Reset Signal for the GLB can come from the Global Reset pin or from a product term within the block. The Output Enable for the I/O cells associated with the GLB comes from a product term within the block. Use of a product term for a control function makes that product term unavailable for use as a logic term. Refer to the following table to determine which logic functions are affected.

There are many additional features in a GLB which allow implementation of logic intensive functions. These features are accessible using the hard Macros from the software and require no intervention on the part of the user.

Figure 6. GLB: Various Logical Combinations



Architectural Description

Product Term Sharing Matrix

Product Term #	Standard Configuration Output Number 3 2 1 0	Four Product Term Bypass Output Number 3 2 1 0	Single Product Term Output Number 3 2 1 0	XOR Function Output Number 3 3 2 2 1 1 0 0	Alternate Function
0	■ ■ ■ ■	■	■	■	
1	■ ■ ■ ■	■		■	
2	■ ■ ■ ■	■		■	
3	■ ■ ■ ■	■		■	
4	■ ■ ■ ■	■	■	■	
5	■ ■ ■ ■	■		■	
6	■ ■ ■ ■	■		■	
7	■ ■ ■ ■	■		■	
8	■ ■ ■ ■	■	■	■	
9	■ ■ ■ ■	■		■	
10	■ ■ ■ ■	■		■	
11	■ ■ ■ ■	■		■	
12	■ ■ ■ ■	■		■	■ CLK/Reset
13	■ ■ ■ ■	■	■	■	
14	■ ■ ■ ■	■		■	
15	■ ■ ■ ■	■		■	
16	■ ■ ■ ■	■		■	
17	■ ■ ■ ■			■	
18	■ ■ ■ ■			■	
19	■ ■ ■ ■			■	■ OE/Reset

This matrix shows how each of the product terms are used in the various modes. As an example, Product Term 12 can be used as an input to the five input OR gate in the standard configuration. This OR gate under standard configuration can be routed to any of the four GLB outputs. Product Term 12 is not used in the four product term bypass mode. When GLB output one is used in the Exclusive OR (XOR) mode Product Term 12 becomes one of the inputs to the four input OR Gate. If Product Term 12 is not used in the logic, then it is available for use as either the Asynchronous Clock signal or the GLB Reset signal.

Architectural Description

The Megablock

The pLSI family is structured into multiple "Megablocks". A Megablock consists of 8 GLBs, an Output Routing Pool (ORP) and 16 I/O Cells. Each of these will be explained in detail in the following sections. These elements are coupled together as shown in figure 7. This logic structure is referred to as the Megablock. The various members of the pLSI family are created by combining from two to six Megablocks on a single device.

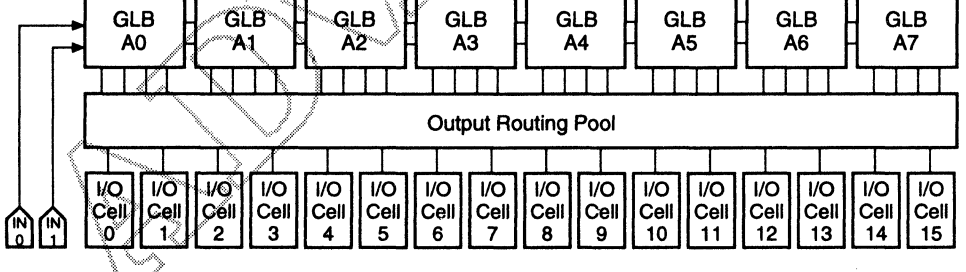
The Megablock shares two sets of resources. The eight GLBs within the Megablock share two dedicated input pins. These dedicated input pins are not available to GLBs in any other Megablock. These pins are dedicated (non-registered) inputs only and are automatically assigned by software. One Output Enable signal is generated within the Megablock and is common to all sixteen of the I/O Cells in the Megablock. The Output Enable signal can be generated using a product term in any of the eight GLBs within the Megablock (see the following section on the Output Enable Multiplexers).

Because of the shared logic within the Megablock, signals which share a common function (counters, busses, etc.) should be grouped within a Megablock. This will allow the user to obtain the best utilization of the logic within the device and eliminate routing bottlenecks.

Megablocks in Each Device

DEVICE	MEGABLOCKS	GLBs	I/O CELLS
pLSI 1016	2	16	32
pLSI 1024	3	24	48
pLSI 1032	4	32	64
pLSI 1048	6	48	96

Figure 7. The Megablock



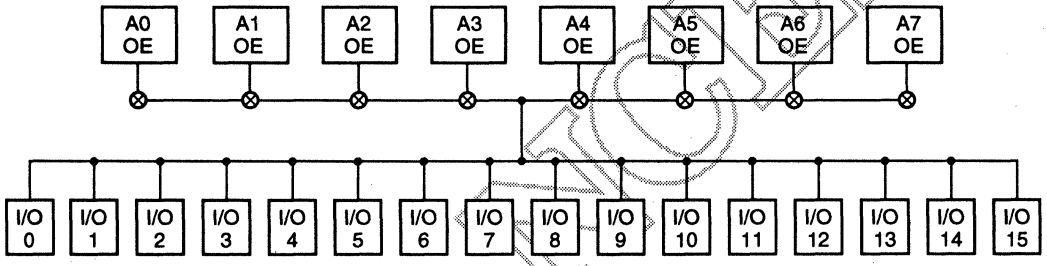
Note: The inputs from the I/O Cell are not shown in this figure, they go directly to the GRP.

Architectural Description

The Output Enable Control

One Output Enable (OE) signal can be generated within each GLB using the OE Product Term 19. One of the eight OE signals from each GLB within a Megablock, is then routed to all of the I/O Cells within the Megablock (see figure 8). This OE signal can simultaneously control all of the 16 I/O cells which are used in 3-state mode. Individual I/O cells also have independent control for permanently enabling or disabling the output buffer (Refer to the I/O Cell Section). Only one OE signal is allowed per Megablock for 3-state operation. The advantage to this approach is that the OE signal can be generated in any GLB within the Megablock which happens to have an unused OE product term. This frees up the other OE product terms for use as logic.

Figure 8. Output Enable Controls



Architectural Description

The Output Routing Pool

The Output Routing Pool (ORP) routes signals from the Generic Logic Block outputs to I/O Cells configured as outputs or bi-directional pins (see figure 9). The purpose of the ORP is to allow greater flexibility when assigning I/O pins. It also simplifies the job for the routing software which results in a higher degree of utilization.

By examining the programmable switch matrix in figure 9, it can be seen that a GLB output can be connected to one of four I/O Cells. Further flexibility is provided by using the PTSA, (figures 3 through 7) which makes the GLB outputs completely interchangeable. This allows the routing program to freely interchange the outputs to achieve the best routability. This is an automatic process and requires no intervention on the part of the user.

The ORP bypass connections (see figure 10) further increase the flexibility of the device. The ORP bypass connect specific GLB outputs to specific I/O Cells at a faster speed. The bypass path tends to restrict the routability of the device and should only be used for critical signals.

Figure 9. Output Routing Pool

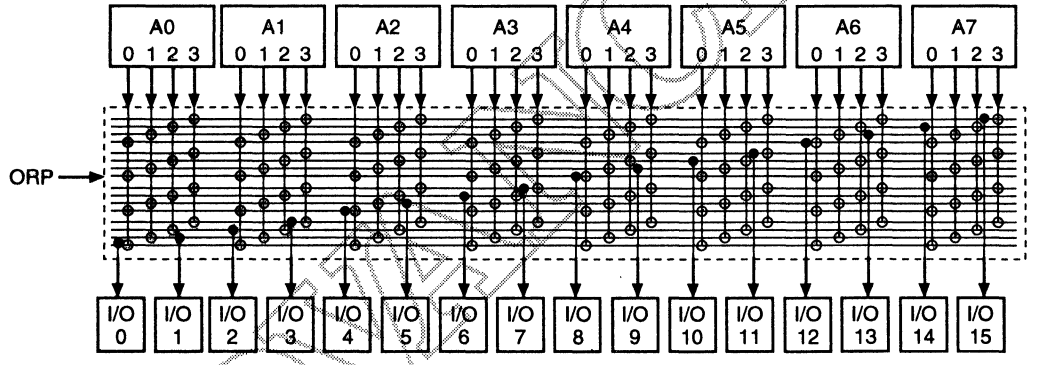
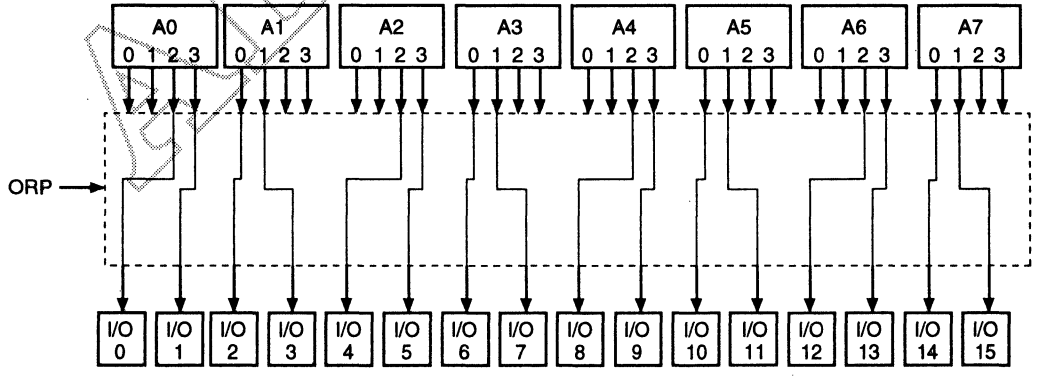


Figure 10. Output Routing Pool showing Bypass



Architectural Description

I/O Cell

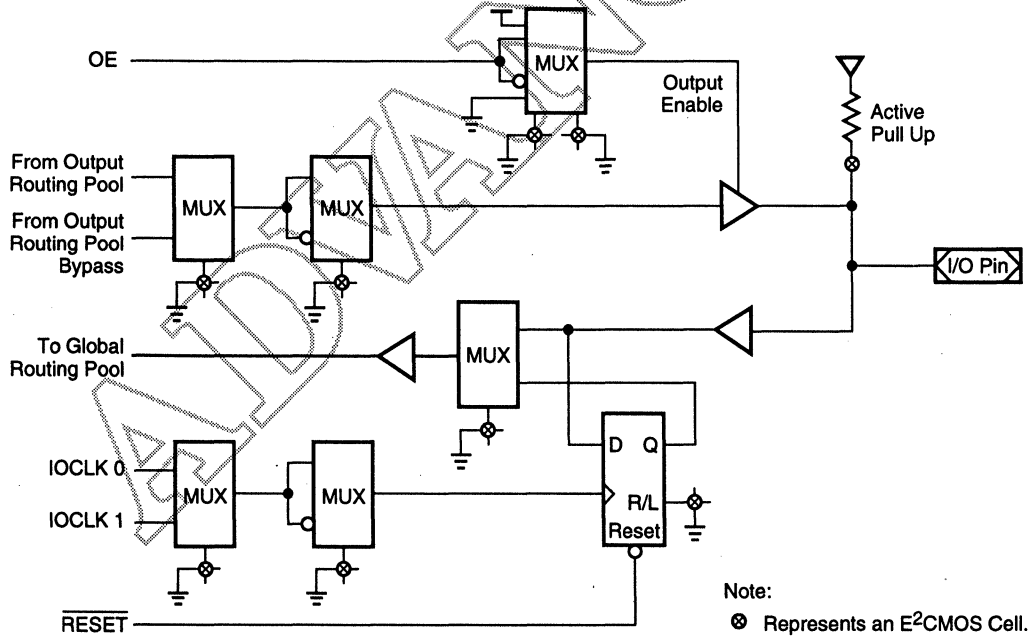
The I/O Cell (see figure 11) is used to route input, output or bidirectional signals connected to the I/O pin. The two logic inputs come from the ORP (see figure 7). One comes from the ORP, and the other comes from the faster ORP bypass. A pair of multiplexers select which signal will be used, and its polarity.

The OE signal comes from the Output Enable. As with the data path, a multiplexer selects the signal polarity. The Output Enable can be set to a logic high (Enabled) when a straight output pin is desired, or logic low (Disabled) when a straight input pin is needed. The Global Reset (RESET) signal is driven by the active low chip reset pin. Each I/O Cell can individually select one of the two clock signals (IOCLK 0 or IOCLK 1). These clock signals are generated by the Clock Distribution Network.

Using the multiplexers, the I/O Cell can be configured as an input, an output, a 3-stated output or a bidirectional I/O. The D-type register can be configured as a level sensitive transparent latch or an edge triggered flip-flop to store the in-coming data. Figure 12 illustrates some of the various I/O cell configurations possible.

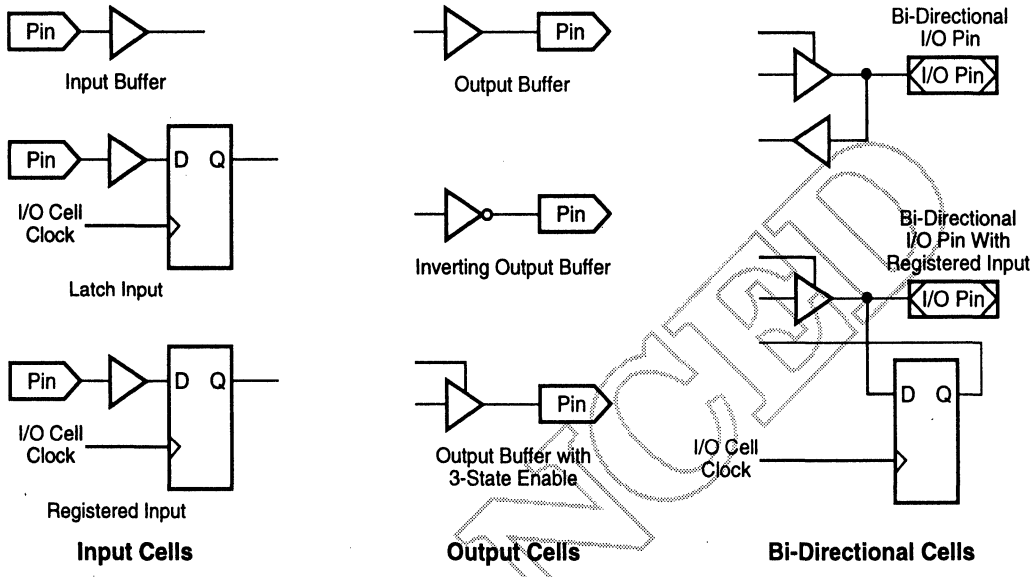
There is an Active Pullup resistor on the I/O pins which is automatically used when the pin is not connected. This prevents the pin from floating and inducing noise into the device or consuming additional power.

Figure 11. I/O Cell Architecture



Architectural Description

Figure 12. Example I/O Cell Configurations



Architectural Description

Clock Distribution Network

The Clock Distribution Network is shown in figure 13. It generates five global clock signals CLK 0, CLK 1, CLK 2 and IOCLK 0, IOCLK 1. The first three, CLK 0, CLK 1 and CLK 2 are used for clocking all the GLBs in the device. Similarly, IOCLK 0 and IOCLK 1 signals are used for clocking all of the I/O Cells in the device. There are four dedicated system clock pins (Y0, Y1, Y2, Y3) which can be directed to any GLB or any I/O Cell using the Clock Distribution Network. The other inputs to the Clock Distribution Network are the 4 outputs of a dedicated clock GLB ("D0" for pLSI 1048). These Clock GLB outputs can be used to create a user-defined internal clocking scheme.

Typically the clock GLB will be clocked using an external main clock pin Y0 connected to global clock signal CLK 0. The outputs of the clock GLB in turn will generate "divide by two" or "divide by four" phases of the CLK 0 which can be connected to CLK 1, CLK 2 or IOCLK 0, IOCLK 1 global clocks.

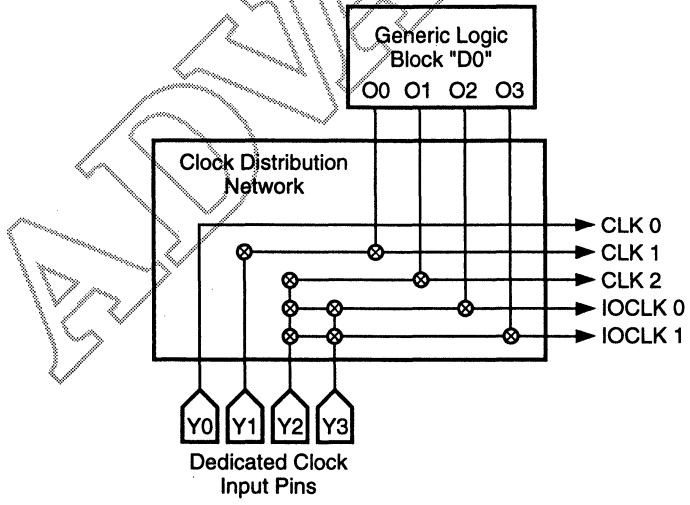
All GLBs also have the capability of generating their own asynchronous clocks using Product Term 12. CLK 0, CLK 1 and CLK 2 feed to their corresponding inputs on all the GLBs (see figure 3).

The two I/O clocks generated in the Clock Distribution Network IOCLK 0 and IOCLK 1, are brought to all 96 of the I/O cells and the user programs the I/O cell to use one of the two.

Global Routing Pool

The Global Routing Pool (GRP) is a Lattice proprietary interconnect structure which offers fast predictable speeds with complete connectivity. The GRP allows the outputs from the GLBs or the I/O cell inputs to be connected to the inputs of the GLBs. Any GLB output is available to the input of all other GLBs, and similarly an input from an I/O pin is available as an input to all of the GLBs. Because of the uniform architecture of the pLSI device, the delays through the Global Routing Pool are both consistent and predictable. However, they are slightly affected by fanout.

Figure 13. Clock Distribution Network



Security Cell

A security cell is provided in the pLSI devices to prevent unauthorized copying of the array patterns. Once programmed, this cell prevents further read access to the functional bits in the device. This cell can only be erased by reprogramming the device, so the original configuration can never be examined once this cell is programmed.

Device Programming

pLSI devices are programmed using a Lattice-approved Device Programmer, available from a number of third party manufacturers. Complete programming of the device takes only a few seconds. Erasing of the device is automatic and is completely transparent to the user. In-system programming is available with ispLSI devices using Lattice programming algorithms.

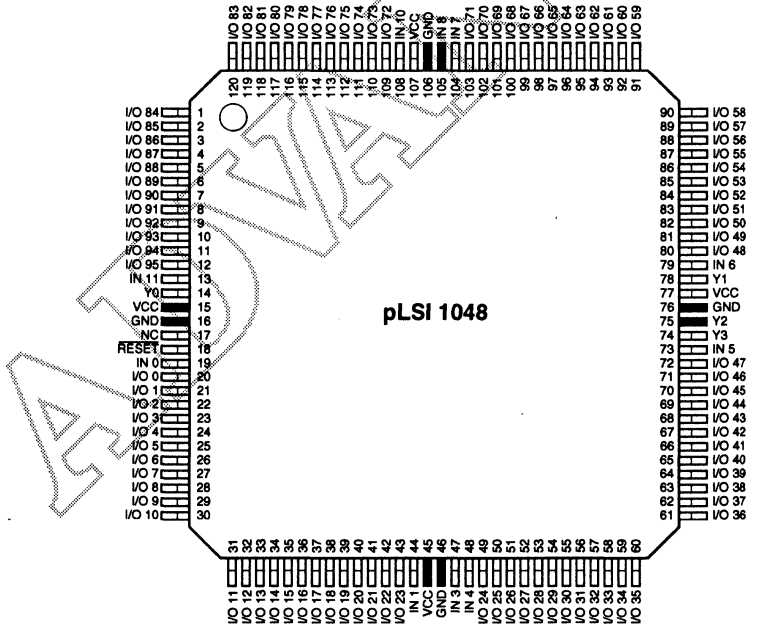
Latch-up Protection

pLSI devices are designed with an on-board charge pump to negatively bias the substrate. The negative bias is of sufficient magnitude to prevent input undershoots from causing the internal circuitry to latchup. Additionally, outputs are designed with n-channel pull-ups instead of the traditional p-channel pull-ups to eliminate any possibility of SCR induced latching.

2

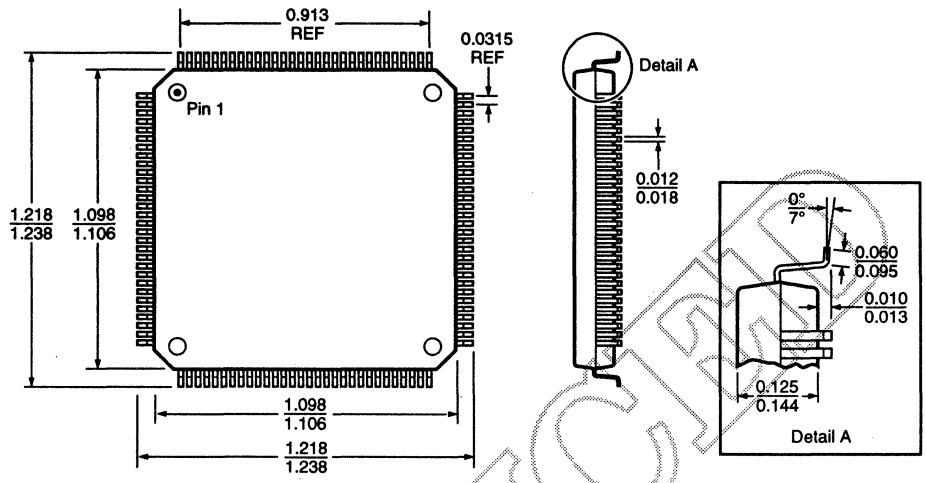
Pin Configuration

pLSI 1048 PQFP Pinout Diagram



120-Pin PQFP

Dimensions in inches MIN./MAX.



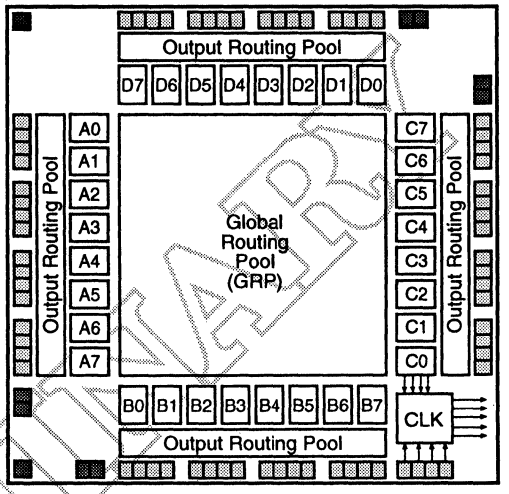
ispLSI™ 1032

in-system programmable Large Scale Integration

Features

- **in-system programmable HIGH DENSITY LOGIC**
  - **— Member of Lattice's ispLSI Family**
  - **— Fully Compatible with Lattice's pLSI™ Family**
  - **— High Speed Global Interconnects**
  - **— 64 I/O Pins, Eight Dedicated Inputs**
  - **— 192 Registers**
  - **— Wide Input Gating for Fast Counters, State Machines, Address Decoders, etc.**
  - **— Small Logic Block Size for Random Logic**
  - **— Security Cell Prevents Unauthorized Copying**
- **HIGH PERFORMANCE E²CMOS® TECHNOLOGY**
  - **— f_{max} = 80 MHz Maximum Operating Frequency**
  - **— t_{pd} = 15 ns Propagation Delay**
  - **— Low Power Consumption (I_{cc} 135mA Typ.)**
  - **— TTL Compatible Inputs and Outputs**
- **in-system programmable 5-VOLT ONLY**
  - **— Change Logic and Interconnects "on-the-fly" in Seconds**
  - **— Reprogram Soldered Device for Debugging**
  - **— Non-Volatile E²CMOS Technology**
  - **— 100% Tested**
- **COMBINES EASE OF USE AND THE FAST SYSTEM SPEED OF PLDs WITH THE DENSITY AND FLEXIBILITY OF FIELD PROGRAMMABLE GATE ARRAYS**
  - **— Complete Programmable Device can Combine Glue Logic and Structured Designs**
  - **— 100% Routable at 80% Utilization**
  - **— Four Dedicated Clock Input Pins**
  - **— Synchronous and Asynchronous Clocks**
  - **— Flexible Pin Placement**
  - **— Optimized Global Routing Pool Allows Global Interconnectivity**
- **pLSI/ispLSI™ DEVELOPMENT SYSTEM (pDS™)**
  - **— Boolean Logic Compiler**
  - **— Automatic Place and Route**
  - **— Manual Partitioning**
  - **— PC Platform**
  - **— Easy to Use Windows Interface**
- **ADVANCED pLSI/ispLSI DEVELOPMENT SYSTEM**
  - **— Industry Standard, Third Party Design Environments**
  - **— Schematic Capture**
  - **— Fully Automatic Partitioning**
  - **— Automatic Place and Route**
  - **— Comprehensive Logic and Timing Simulation**
  - **— PC and Workstation Platforms**

Functional Block Diagram



Description

The Lattice ispLSI 1032 is a High Density Programmable Logic Device featuring 5-Volt in-system programmability and in-system diagnostic capabilities. The device contains 192 Registers, 64 Universal I/O pins, eight Dedicated Input Pins, four Dedicated Clock Input Pins and a Global Routing Pool (GRP). The GRP provides complete interconnectivity between all of these elements. It is the first device which offers non-volatile "on-the-fly" reprogrammability of the logic, as well as the interconnects to provide truly reconfigurable systems. It is architecturally and parametrically compatible to the pLSI 1032 device, but multiplexes four of the dedicated input pins to control in-system programming.

The basic unit of logic on the ispLSI 1032 device is the Generic Logic Block (GLB). The GLBs are labeled A0, A1 .. D7 (see figure 1). There are a total of 32 GLBs in the ispLSI 1032 device. Each GLB has 18 inputs, a programmable AND/OR/Exclusive OR array, and four outputs which can be configured to be either combinatorial or registered. Inputs to the GLB come from the GRP. All of the GLB outputs are brought back into the GRP so that they can be connected to the inputs of any other GLB on the device.

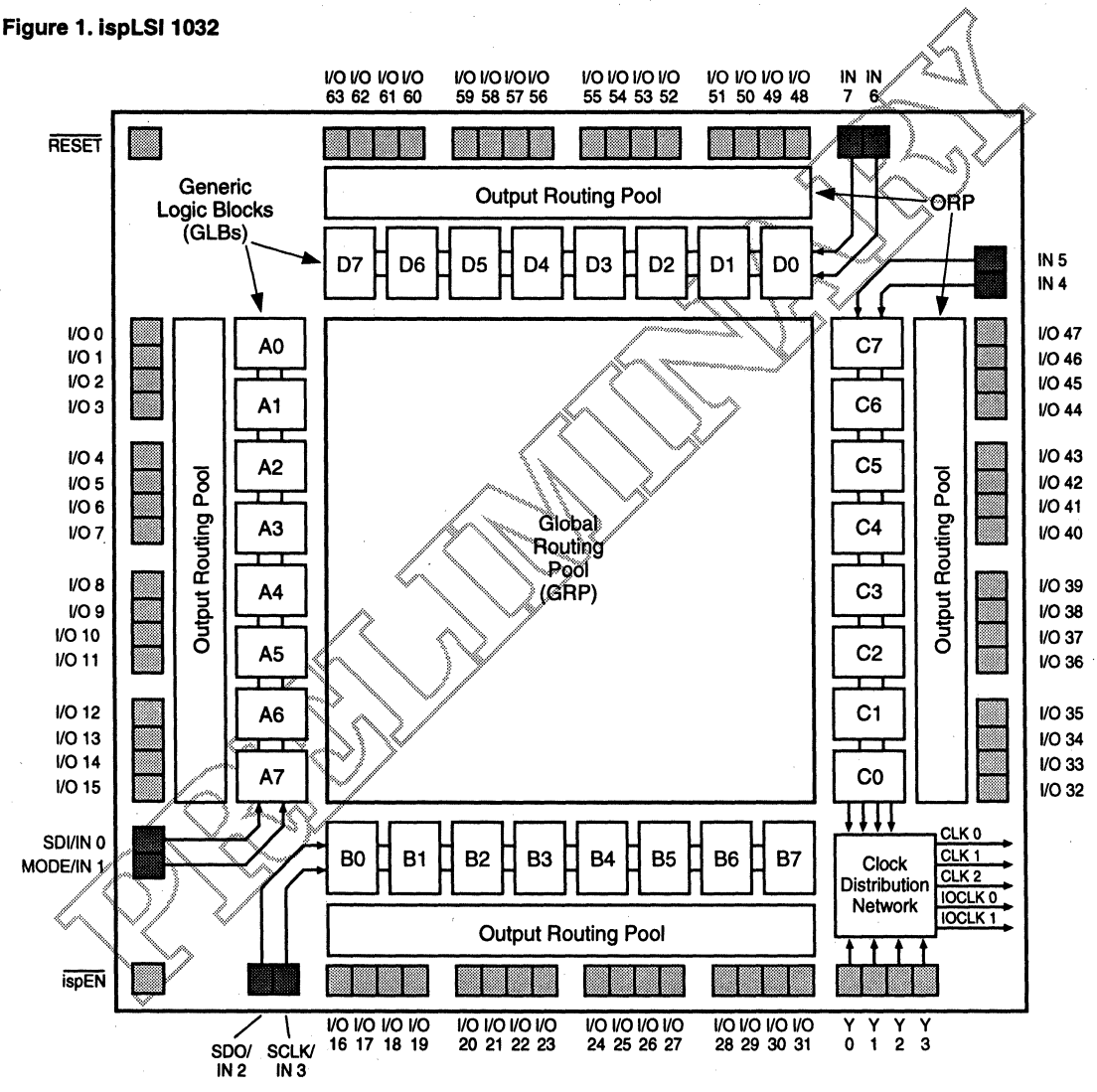
LATTICE SEMICONDUCTOR CORP., 5555 Northeast Moore Ct., Hillsboro, Oregon 97124, U.S.A.
Tel. 1-800-LATTICE (528-8423); FAX (503) 681-3037

January 1992. Rev. A

The device also has 64 I/O Cells, each of which is directly connected to an I/O pin. Each I/O cell can be individually programmed to be a combinatorial input, registered input, latched input, output or bi-directional I/O pin with 3-state. The signal levels are TTL compatible voltages and the output drivers can source 4 mA or sink 8 mA.

Functional Block Diagram

Figure 1. ispLSI 1032



Description (continued)

The 64 I/O Cells are grouped into four sets of 16 each. as shown in figure 1. Each of these I/O groups is associated with a logic Megablock through the use of the Output Routing Pool (ORP) and shares a common Output Enable (OE) signal.

Eight GLBs, 16 I/O Cells and one ORP are connected together to make a logic Megablock. The Megablock is defined by the resources that it shares. The outputs of the eight GLBs are connected to a set of 16 universal I/O cells by the ORP. The ispLSI 1032 Device contains four of these Megablocks.

The GRP has as its inputs the outputs from all of the GLBs and all of the inputs from the bi-directional I/O cells. All of these signals are made available to the inputs of the GLBs. Delays through the GRP have been equalized to minimize timing skew and logic glitching.

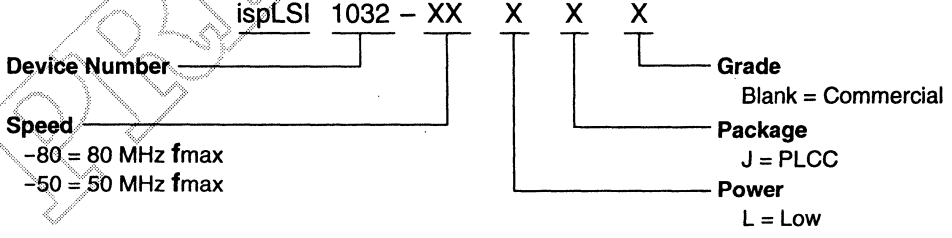
Clocks in the ispLSI 1032 device are selected using the Clock Distribution Network. Four dedicated clock pins (Y0 to Y3) are brought into the distribution network, and five outputs (CLK 0 to CLK 2 and IOCLK 0, IOCLK 1) are provided to route clocks to the GLBs and I/O cells. The Clock Distribution Network can also be driven from a special GLB (C0 on the ispLSI 1032 device). The logic of this GLB allows the user to create an internal clock from a combination of internal signals within the device.

The ispLSI 1032 device is part of Lattice's in-system programmable Large Scale Integration (ispLSI) family. This family contains a range of devices from the ispLSI 1016, with 96 registers, to the ispLSI 1048 with 288 registers. The ispLSI Family Product Selector Guide below lists key attributes of the devices along with the number of resources available.

ispLSI Family Product Selector Guide

DEVICE	ispLSI 1016	ispLSI 1024	ispLSI 1032	ispLSI 1048
GLBs	16	24	32	48
Registers	96	144	192	288
I/O Pins	32	48	64	96
Dedicated Inputs	4	6	8	10
Pin Count	44	68	84	120

Ordering Information



Pin Description

Name	PLCC Pin Numbers	Description
I/O 0 - I/O 3 I/O 4 - I/O 7 I/O 8 - I/O 11 I/O 12 - I/O 15 I/O 16 - I/O 19 I/O 20 - I/O 23 I/O 24 - I/O 27 I/O 28 - I/O 31 I/O 32 - I/O 35 I/O 36 - I/O 39 I/O 40 - I/O 43 I/O 44 - I/O 47 I/O 48 - I/O 51 I/O 52 - I/O 55 I/O 56 - I/O 59 I/O 60 - I/O 63	26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18	Input/Output Pins - These are the general purpose I/O pins used by the logic array.
IN 4 - IN 7	67, 84, 2, 19	Dedicated input pins to the device.
$\overline{\text{ispEN}}$	23	Input – Dedicated in-system programming enable input pin. This pin is brought low to enable the programming mode. The MODE, SDI, SDO and SCLK options become active.
SDI/IN 0	25	Input – This pin performs two functions. It is a dedicated input pin when $\overline{\text{ispEN}}$ is logic high. When $\overline{\text{ispEN}}$ is logic low, it functions as an input pin to load programming data into the device. SDI/IN 0 also is used as one of the two control pins for the isp state machine.
MODE/IN 1	42	Input – This pin performs two functions. It is a dedicated input pin when $\overline{\text{ispEN}}$ is logic high. When $\overline{\text{ispEN}}$ is logic low, it functions as a pin to control the operation of the isp state machine.
SDO/IN 2	44	Input/Output – This pin performs two functions. It is a dedicated clock input pin when $\overline{\text{ispEN}}$ is logic high. When $\overline{\text{ispEN}}$ is logic low, it functions as an output pin to read serial shift register data.
SCLK/IN 3	61	Input – This pin performs two functions. It is a dedicated input when $\overline{\text{ispEN}}$ is logic high. When $\overline{\text{ispEN}}$ is logic low, it functions as a clock pin for the Serial Shift Register.
$\overline{\text{RESET}}$	24	Active Low (0) Reset pin which resets all of the GLB and I/O registers in the device.
Y0	20	Dedicated Clock input. This clock input is connected to one of the clock inputs of all of the GLBs on the device.
Y1	66	Dedicated Clock input. This clock input is brought into the clock distribution network, and can optionally be routed to any GLB on the device.
Y2	63	Dedicated Clock input. This clock input is brought into the clock distribution network, and can optionally be routed to any GLB and/or any I/O Cell on the device.
Y3	62	Dedicated Clock input. This clock input is brought into the clock distribution network, and can optionally be routed to any I/O Cell on the device.
GND	1, 22, 43, 64	Ground (GND)
V_{CC}	21, 65	V_{CC}

Absolute Maximum Ratings [1]

Supply Voltage V_{CC} . . . . . -0.5 to +7.0V

Input Voltage Applied. . . . . -2.5 to V_{CC} +1.0V

Off-State Output Voltage Applied . . . . . -2.5 to V_{CC} +1.0V

Storage Temperature . . . . . -65 to 125°C

Ambient Temp. with Power Applied . . . . . -55 to 125°C

1. Stresses above those listed under the "Absolute Maximum Ratings" may cause permanent damage to the device. Functional operation of the device at these or at any other conditions above those indicated in the operational sections of this specification is not implied (while programming, follow the programming specifications).

DC Recommended Operating Condition

SYMBOL	PARAMETER	MIN.	MAX.	UNITS
T_A	Ambient Temperature	0	70	°C
V_{CC}	Supply Voltage	4.75	5.25	V
V_{IL}	Input Low Voltage	0	0.8	V
V_{IH}	Input High Voltage	2.0	V_{CC}	V

Capacitance (T_A=25°C,f=1.0 MHz)

SYMBOL	PARAMETER	MAXIMUM[1]	UNITS	TEST CONDITIONS
C_1	Input Capacitance	8	pf	V_{CC}=5.0V, V_{IN}=2.0V
C_2	I/O, Y Capacitance	10	pf	V_{CC}=5.0V, VI/O, Y=2.0V

1. Guaranteed but not 100% tested.

Data Retention Specifications

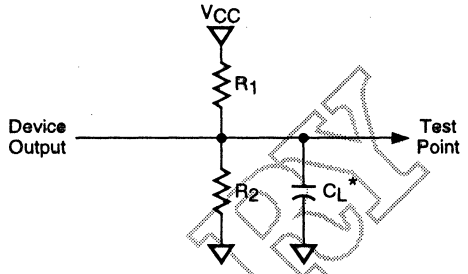
PARAMETER	MINIMUM	MAXIMUM	UNITS
Data Retention	20	–	YEARS
Erase/Reprogram Cycles	–	1000	CYCLES

Switching Test Conditions

Input Pulse Levels	GND to 3.0V
Input Rise and Fall Times	3ns 10% to 90%
Input Timing Reference Levels	1.5V
Output Timing Reference Levels	1.5V
Output Load	See Figure 2

3-state levels are measured 0.5V from steady-state active level.

Figure 2. Test Load



*C_L Indicates Test Fixture and Probe Total Capacitance

Output Load Conditions (see figure 2)

Test Condition		R1	R2	CL
1		470Ω	390Ω	35pF
2	Active High	∞	390Ω	35pF
	Active Low	470Ω	390Ω	35pF
3	Active High to Z at V_{OH} - 0.5V	∞	390Ω	5pF
	Active Low to Z at V_{OL} + 0.5V	470Ω	390Ω	5pF

DC Electrical Characteristics

Over Recommended Operating Conditions

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNITS
V_{OL}	Output Low Voltage	I_{OL} =8 mA.	–	–	0.4	V
V_{OH}	Output High Voltage	I_{OH} =-4 mA.	2.4	–	–	V
I_{IL}	Input or I/O Low Leakage Current	$0V \le V_{IN} \le V_{IL}$ (MAX.)	–	–	-10	µA
I_{IH}	Input or I/O High Leakage Current	$V_{IH} \le V_{IN} \le V_{CC}$	–	–	10	µA
I_{OS}[1]	Output Short Circuit Current	V_{CC} = 5V, V_{OUT} = 0.5V	-60	–	-200	mA
I_{CC}[2]	Operating Power Supply Current	V_{IL} = 0.5V, V_{IH} = 3.0V F_{TOGGLE} = 20 MHz	–	135	195	mA

1. One output at a time for a maximum duration of one second. (Vout = 0.5V)
2. Measured at a frequency of 20 MHz using eight 16-bit counters.

External Switching Characteristics [1, 2, 3] ispLSI 1032-80

Over Recommended Operating Conditions

PARAMETER	TEST [6] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
t_{pd1}	1	1	Data Propagation Delay, 4PT bypass, ORP bypass	–	12	15	ns
t_{pd2}	1	2	Data Propagation Delay	–	15	20	ns
t_{co1} [5]	1	3	External Clock to Output Delay, ORP bypass	–	8	11	ns
t_{co2} [5]	1	4	External Clock to Output Delay	–	9	14	ns
t_{co3}	1	5	Internal Synch. Clock to Output Delay	–	15	20	ns
t_{co4}	1	6	Asynchronous Clock to Output Delay	–	13	20	ns
t_{r1}	1	7	External Pin Reset to Output Delay	–	13	20	ns
t_{r2}	1	8	Asynchronous PT Reset to Output Delay	–	15	22	ns
t_{en}	2	9	Input to Output Enable	–	13	20	ns
t_{dis}	3	10	Input to Output Disable	–	13	20	ns

External AC Recommended Operating Conditions [1, 2, 3] ispLSI 1032-80

Over Recommended Operating Conditions

PARAMETER	TEST [6] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
f_{max} [4]	1	11	Clock Frequency with Internal Feedback	–	100	80	MHz
f_{max} (External)	1	12	Clock Frequency with External Feedback	–	70	50	MHz
t_{su1}	–	13	Setup Time before External Synch. Clock, 4PT bypass	9	6	–	ns
t_{su2}	–	14	Setup Time before External Synch Clock	12	8	–	ns
t_{su3}	–	15	Setup Time before Internal Synch. Clock	9	3	–	ns
t_{su4}	–	16	Setup Time before Asynchronous Clock	9	4	–	ns
t_{h1}	–	17	Hold time after External Synchronous Clock, 4PT bypass	2	-1	–	ns
t_{h2}	–	18	Hold time after External Synchronous Clock	2	-1	–	ns
t_{h3}	–	19	Hold time after Internal Synchronous Clock	8	2	–	ns
t_{h4}	–	20	Hold time after Asynchronous Clock	8	1	–	ns
t_{rw1}	–	21	External Reset Pulse Duration	10	8	–	ns
t_{rw2}	–	22	Asynchronous Reset Pulse Duration	10	8	–	ns
t_{wh1}, t_{wl1}	–	23,24	External Synchronous Clock Pulse Duration, High, Low	6	5	–	ns
t_{wh2}, t_{wl2}	–	25,26	Asynchronous Clock Pulse Duration, High, Low	6	5	–	ns

1. External Parameters are tested and guaranteed.
2. See Timing Technical Note for further details.
3. Unless noted, all parameters use ORP, GRP fanout of four, PTSA, and are measured with 16 outputs switching.
4. Standard 16-bit counter implementation using GRP feedback.
5. Clock to output specifications include a maximum skew of 2 ns.
6. Refer to Switching Test Conditions section.

Switching Characteristics[1, 2, 3] — ispLSI 1032-80

Using I/O Cell

PARAMETER	TEST[4] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
t_{su5}	–	27	Setup Time before External Synchronous Clock	5	0	–	ns
t_{su6}	–	28	Setup Time before Internal Synchronous Clock	0	-3	–	ns
t_{h5}	–	29	Hold Time after External Synchronous Clock	8	4	–	ns
t_{h6}	–	30	Hold Time after Internal Synchronous Clock	15	11	–	ns
t_{wh3}, t_{wl3}	–	31,32	Clock Pulse Duration, High, Low	6	5	–	ns

1. External Parameters are tested and guaranteed.
2. See Timing Technical Note for further details.
3. Unless noted, all parameters use ORP, GRP fanout of four, PTSA, and are measured with 16 outputs switching.
4. Refer to Switching Test Conditions section.

External Switching Characteristics[1, 2, 3] ispLSI 1032-50

Over Recommended Operating Conditions

PARAMETER	TEST[6] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
t_{pd1}	1	1	Data Propagation Delay, 4PT bypass, ORP bypass	–	16	20	ns
t_{pd2}	1	2	Data Propagation Delay, ORP	–	19	25	ns
t_{co1}[5]	1	3	External Clock to Output Delay, ORP bypass	–	12	16	ns
t_{co2}[5]	1	4	External Clock to Output Delay	–	15	20	ns
t_{co3}	1	5	Internal Synch. Clock to Output Delay	–	21	28	ns
t_{co4}	1	6	Asynchronous Clock to Output Delay	–	21	28	ns
t_{r1}	–	7	External Pin Reset to Output Delay	–	21	28	ns
t_{r2}	–	8	Asynchronous PT Reset to Output Delay	–	24	30	ns
t_{en}	2	9	Input to Output Enable	–	21	28	ns
t_{dis}	3	10	Input to Output Disable	–	21	28	ns

External AC Recommended Operating Conditions[1, 2, 3] ispLSI 1032-50

Over Recommended Operating Conditions

PARAMETER	TEST[6] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
f_{max}[4]	1	11	Clock Frequency with Internal Feedback	–	70	50	MHz
f_{max} (External)	1	12	Clock Frequency with External Feedback	–	45	33	MHz
t_{su1}	–	13	Setup Time before External Synch. Clock, 4PT bypass	14	10	–	ns
t_{su2}	–	14	Setup Time before External Synch Clock	17	13	–	ns
t_{su3}	–	15	Setup Time before Internal Synch. Clock	13	9	–	ns
t_{su4}	–	16	Setup Time before Asynchronous Clock	13	9	–	ns
t_{h1}	–	17	Hold time after External Synchronous Clock, 4PT bypass	7	3	–	ns
t_{h2}	–	18	Hold time after External Synchronous Clock	7	3	–	ns
t_{h3}	–	19	Hold time after Internal Synchronous Clock	11	5	–	ns
t_{h4}	–	20	Hold time after Asynchronous Clock	11	5	–	ns
t_{rw1}	–	21	External Reset Pulse Duration	15	13	–	ns
t_{rw2}	–	22	Asynchronous Reset Pulse Duration	15	13	–	ns
t_{wh1}, t_{wl1}	–	23,24	External Synchronous Clock Pulse Duration, High, Low	10	8	–	ns
t_{wh2}, t_{wl2}	–	25,26	Asynchronous Clock Pulse Duration, High, Low	10	8	–	ns

1. External Parameters are tested and guaranteed.
2. See Timing Technical Note for further details.
3. Unless noted, all parameters use ORP, GRP fanout of four, PTSA, and are measured with 16 outputs switching.
4. Standard 16-bit counter implementation using GRP feedback.
5. Clock to output specifications include a maximum skew of 2 ns.
6. Refer to Switching Test Conditions section.

Switching Characteristics[1, 2, 3] — ispLSI 1032-50

Using I/O Cell

PARAMETER	TEST[4] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
t_{su5}	–	27	Setup Time before External Synchronous Clock	10	5	–	ns
t_{su6}	–	28	Setup Time before Internal Synchronous Clock	0	-5	–	ns
t_{h5}	–	29	Hold Time after External Synchronous Clock	12	6	–	ns
t_{h6}	–	30	Hold Time after Internal Synchronous Clock	20	15	–	ns
t_{wh3}, t_{wl3}	–	31,32	Clock Pulse Duration, High, Low	10	8	–	ns

1. External Parameters are tested and guaranteed.
2. See Timing Technical Note for further details.
3. Unless noted, all parameters use ORP, GRP fanout of four, PTSA, and are measured with 16 outputs switching.
4. Refer to Switching Test Conditions section.

Architectural Description

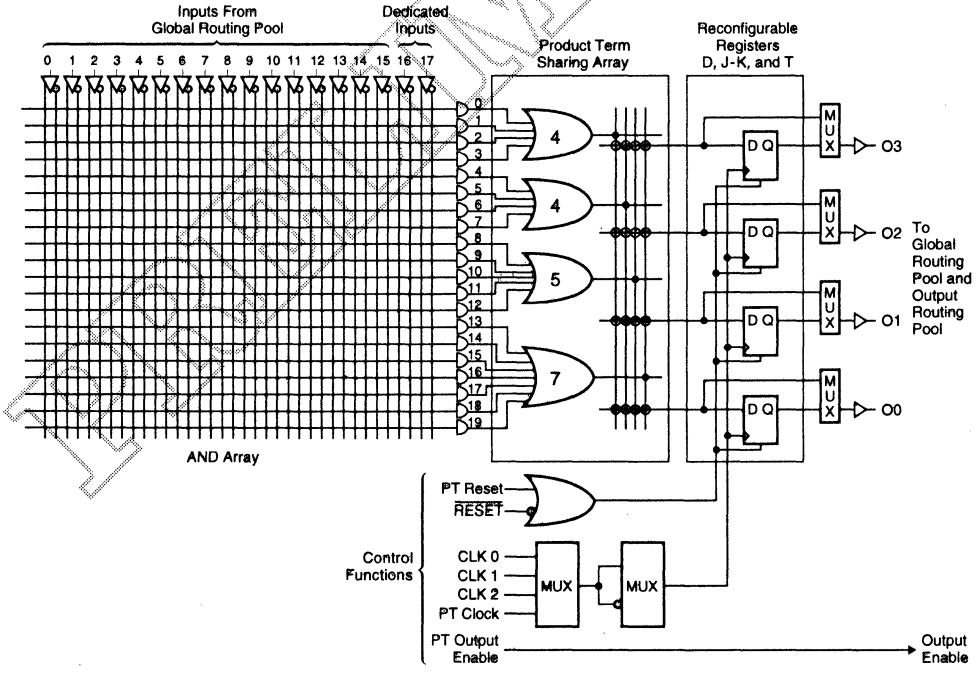
The Generic Logic Block

The Generic Logic Block (GLB) is the standard logic block of the Lattice High Density ispLSI Device. This GLB has 18 inputs, four outputs and the logic necessary to implement most standard logic functions. The internal logic of the GLB is divided into four separate sections (see figure 3). The AND Array, the Product Term Sharing Array (PTSA), the Reconfigurable Registers, and the Control Functions. The AND array consists of 20 product terms which can produce the logical sum of any of the 18 GLB inputs. Sixteen of the inputs come from the Global Routing Pool, and are either feedback signals from any of the 32 GLBs or inputs from the external I/O Cells. The two remaining inputs come directly from two dedicated input pins. These signals are available to the product terms in both the logical true and the complemented forms which makes boolean logic reduction easier.

The PTSA takes the 20 product terms and allocates them to the four GLB outputs. There are four OR gates, with four, four, five and seven product terms (see figure 3). The output of any of these gates can be routed to any of the four GLB outputs, and if more product terms are needed, the PTSA can combine them as necessary. If the users' main concern is speed, the PTSA can use a bypass circuit with four product terms to increase the performance of the cell (see figure 4). This can be done to any or all of the four outputs from the GLB.

The Reconfigurable Registers consist of four D-type flip-flops with an Exclusive OR Gate on the input. The Exclusive OR gate in the GLB can be used either as a logic element or to reconfigure the D-type flip-flop to emulate a J-K, or T-type flip-flop (see figure 5). This greatly simplifies the design of counters, comparators and ALU type functions. The registers can be bypassed if the user needs a combinatorial output. Each register output is brought back into the Global Routing Pool and is also brought to the I/O cells via the Output Routing Pool. Reconfigurable registers are not available when the four product term bypass is used.

Figure 3. GLB: Product Term Sharing Array



Architectural Description

Figure 4. GLB: Four Product Term Bypass

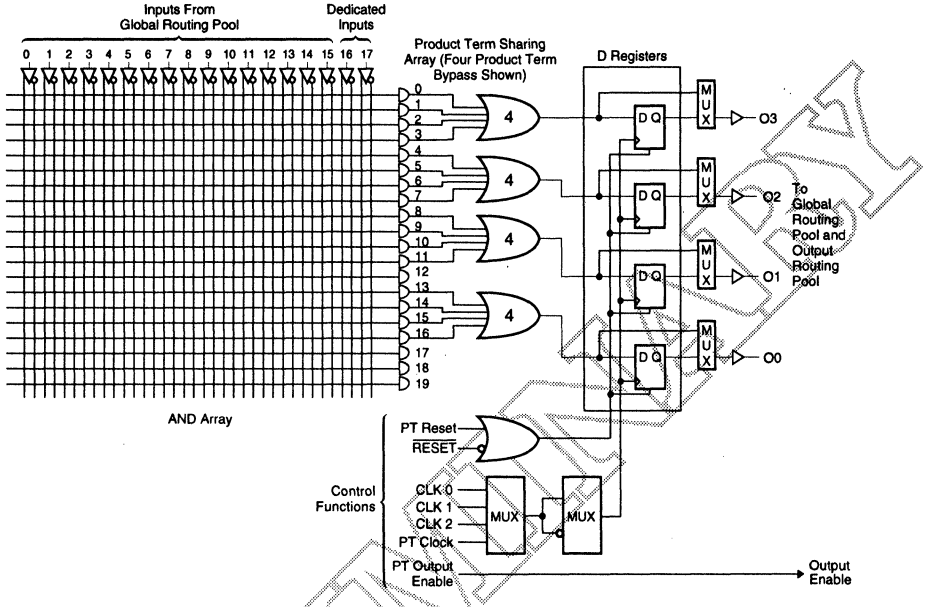
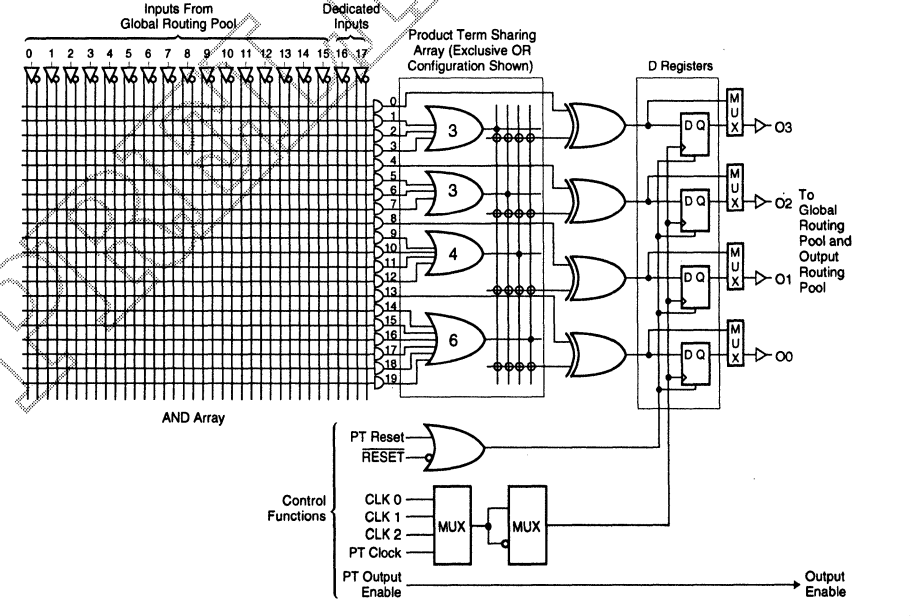


Figure 5. GLB: Exclusive OR Gate



Architectural Description

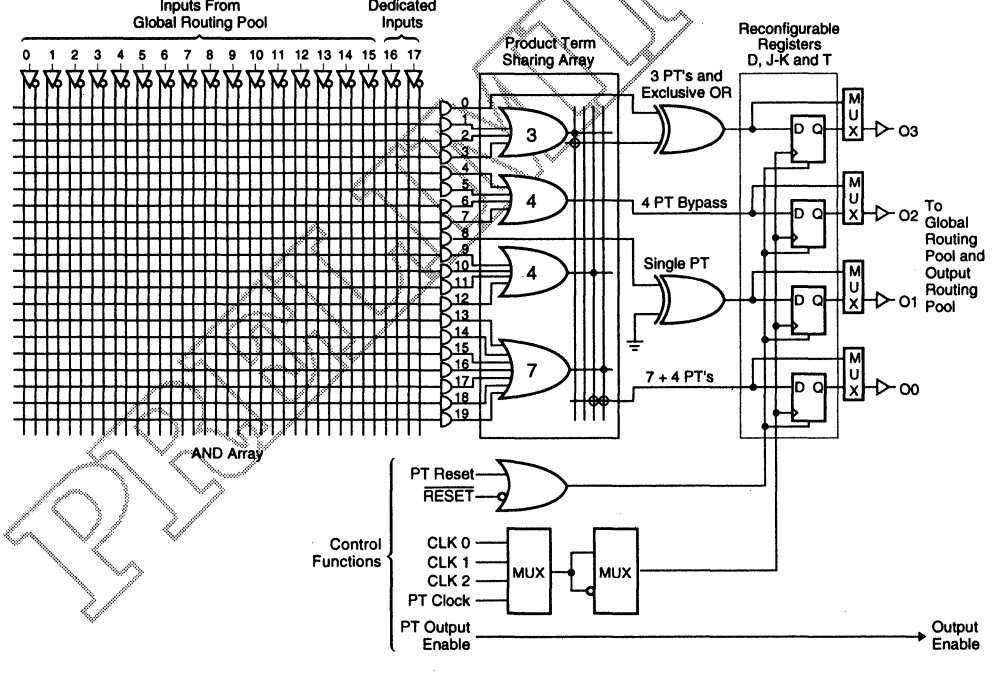
The Generic Logic Block (continued)

The PTSA is flexible enough to allow these features to be used in virtually any combination that the user desires. In the GLB shown in figure 6, Output Three (O3) is configured using the XOR Gate and Output Two (O2) is configured using the four Product Term Bypass. Output One (O1) uses one of the inputs from the five Product Term OR gate while Output Zero (O0) combines the remaining four product terms with all of the product terms from the seven Product Term OR gate for a total of eleven (7+4).

Various signals which control the operation of the GLB are developed in the Control Function section. The clock for the registers can come from any of three sources developed in the Clock Distribution Network (See Clock Distribution Network Section) or from a product term within the GLB. The Reset Signal for the GLB can come from the Global Reset pin or from a product term within the block. The Output Enable for the I/O cells associated with the GLB comes from a product term within the block. Use of a product term for a control function makes that product term unavailable for use as a logic term. Refer to the following table to determine which logic functions are affected.

There are additional features in a GLB which allow implementation of logic intensive functions. These features are accessible using the hard Macros from the software and require no intervention on the part of the user.

Figure 6. GLB: Various Logical Combinations



Architectural Description

Product Term Sharing Matrix

Product Term #	Standard Configuration Output Number 3 2 1 0	Four Product Term Bypass Output Number 3 2 1 0	Single Product Term Output Number 3 2 1 0	XOR Function Output Number 3 3 2 2 1 1 0 0	Alternate Function
0	■ ■ ■ ■	■	■	■	
1	■ ■ ■ ■	■		■	
2	■ ■ ■ ■	■		■	
3	■ ■ ■ ■	■		■	
4	■ ■ ■ ■	■	■	■	
5	■ ■ ■ ■	■		■	
6	■ ■ ■ ■	■		■	
7	■ ■ ■ ■	■		■	
8	■ ■ ■ ■	■	■	■	
9	■ ■ ■ ■	■		■	
10	■ ■ ■ ■	■		■	
11	■ ■ ■ ■	■		■	
12	■ ■ ■ ■	■		■	■ CLK/Reset
13	■ ■ ■ ■	■	■	■	
14	■ ■ ■ ■	■		■	
15	■ ■ ■ ■	■		■	
16	■ ■ ■ ■	■		■	
17	■ ■ ■ ■			■	
18	■ ■ ■ ■			■	
19	■ ■ ■ ■			■	■ OE/Reset

This matrix shows how each of the product terms are used in the various modes. As an example, Product Term 12 can be used as an input to the five input OR gate in the standard configuration. This OR gate under standard configuration can be routed to any of the four GLB outputs. Product Term 12 is not used in the four product term bypass mode. When GLB output one is used in the Exclusive OR (XOR) mode Product Term 12 becomes one of the inputs to the four input OR Gate. If Product Term 12 is not used in the logic, then it is available for use as either the Asynchronous Clock signal or the GLB Reset signal.

Architectural Description

The Megablock

The ispLSI family is structured into multiple "Megablocks". A Megablock consists of 8 GLBs, an Output Routing Pool (ORP) and 16 I/O Cells. Each of these will be explained in detail in the following sections. These elements are coupled together as shown in figure 7. This logic structure is referred to as the Megablock. The various members of the ispLSI family are created by combining from two to six Megablocks on a single device.

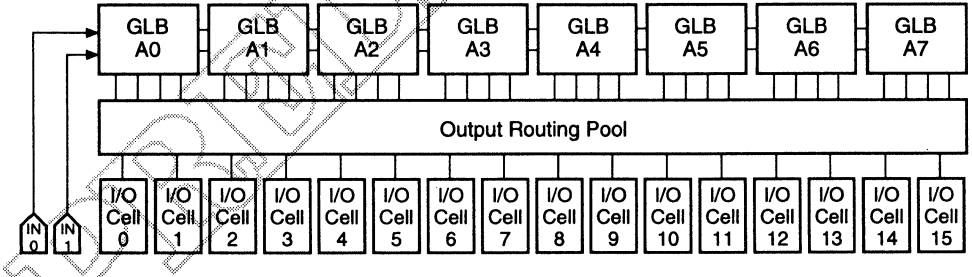
The Megablock shares two sets of resources. The eight GLBs within the Megablock share two dedicated input pins. These dedicated input pins are not available to GLBs in any other Megablock. These pins are dedicated (non-registered) inputs only and are automatically assigned by software. One Output Enable signal is generated within the Megablock and is common to all sixteen of the I/O Cells in the Megablock. The Output Enable signal can be generated using a product term in any of the eight GLBs within the Megablock (See the following section on the Output Enable Multiplexers).

Because of the shared logic within the Megablock, signals which share a common function (counters, busses, etc.) should be grouped within a Megablock. This will allow the user to obtain the best utilization of the logic within the device and eliminate routing bottlenecks.

Megablocks in Each Device

DEVICE	MEGABLOCKS	GLBs	I/O CELLS
ispLSI 1016	2	16	32
ispLSI 1024	3	24	48
ispLSI 1032	4	32	64
ispLSI 1048	6	48	96

Figure 7. The Megablock



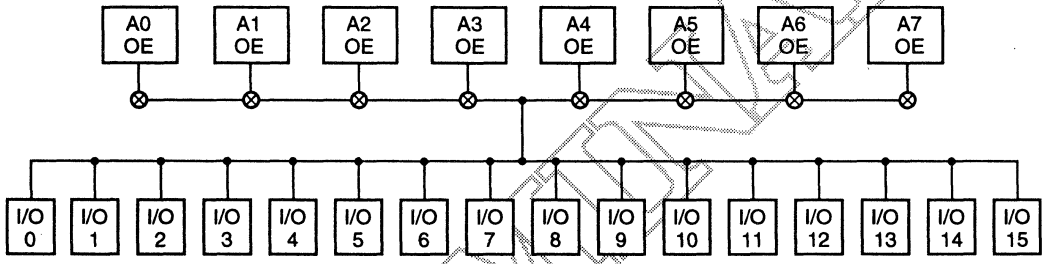
Note: The inputs from the I/O Cell are not shown in this figure, they go directly to the GRP.

Architectural Description

The Output Enable Control

One Output Enable (OE) signal can be generated within each GLB using the OE Product Term 19. One of the eight OE signals from each GLB within a Megablock, is then routed to all of the I/O Cells within the Megablock (see figure 8). This OE signal can simultaneously control all of the 16 I/O cells which are used in 3-state mode. Individual I/O cells also have independent control for permanently enabling or disabling the output buffer (Refer to the I/O Cell Section). Only one OE signal is allowed per Megablock for 3-state operation. The advantage to this approach is that the OE signal can be generated in any GLB within the Megablock which happens to have an unused OE product term. This frees up the other OE product terms for use as logic.

Figure 8. Output Enable Controls



Architectural Description

The Output Routing Pool

The Output Routing Pool (ORP) routes signals from the Generic Logic Block outputs to I/O Cells configured as outputs or bi-directional pins (see figure 9). The purpose of the ORP is to allow greater flexibility when assigning I/O pins. It also simplifies the job for the routing software which results in a higher degree of utilization.

By examining the programmable switch matrix in figure 9, it can be seen that a GLB output can be connected to one of four I/O Cells. Further flexibility is provided by using the PTSA, (Figures 3 through 7) which makes the GLB outputs completely interchangeable. This allows the routing program to freely interchange the outputs to achieve the best routability. This is an automatic process and requires no intervention on the part of the user.

The ORP bypass connections (see figure 10) further increase the flexibility of the device. The ORP bypass connect specific GLB outputs to specific I/O Cells at a faster speed. The bypass path tends to restrict the routability of the device and should only be used for critical signals.

Figure 9. Output Routing Pool

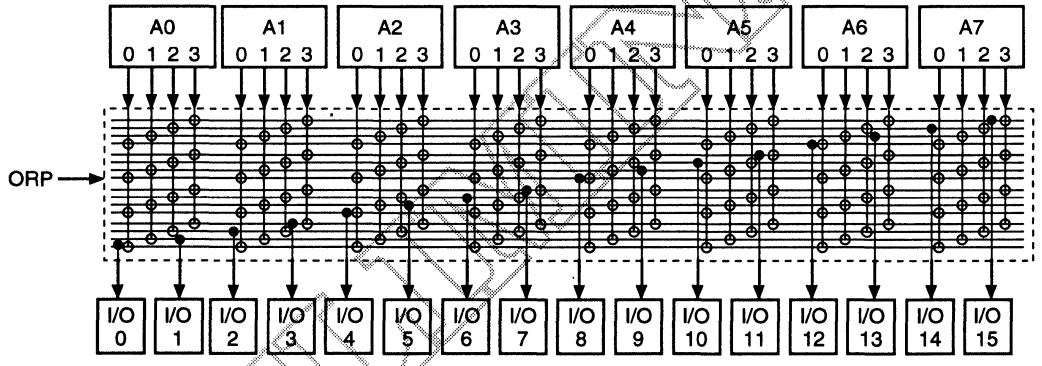
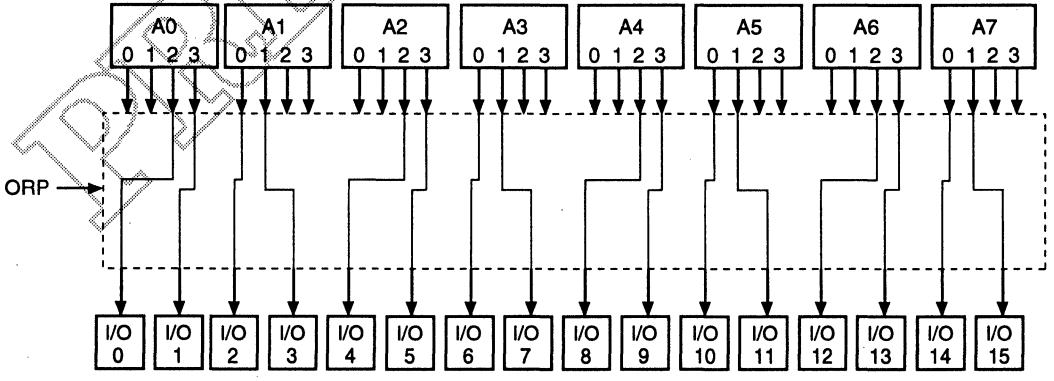


Figure 10. Output Routing Pool showing Bypass



Architectural Description

I/O Cell

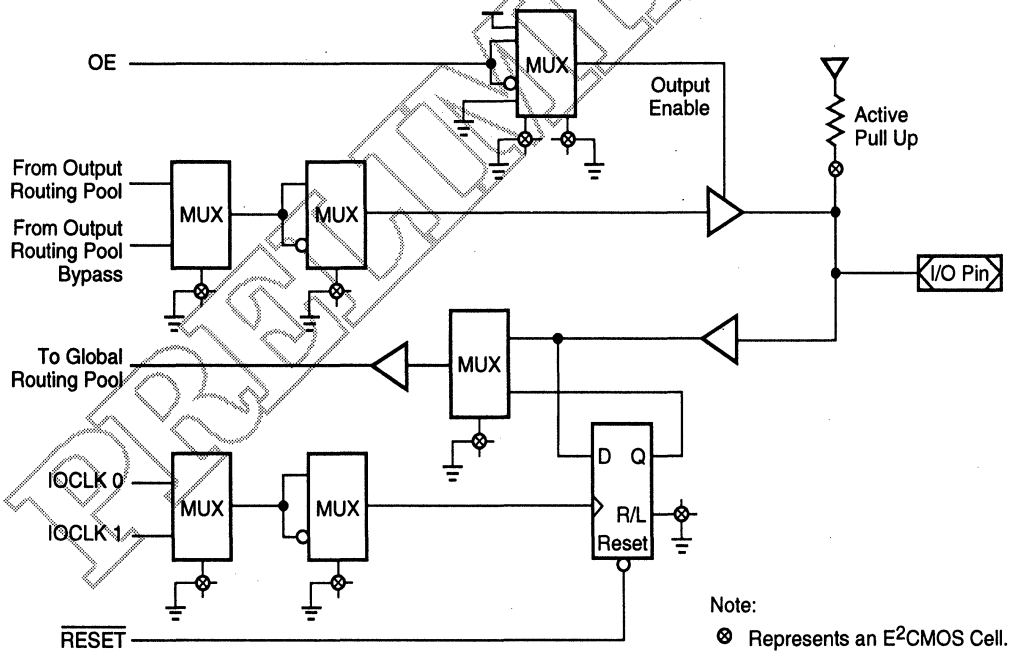
The I/O Cell (see figure 11) is used to route input, output or bidirectional signals connected to the I/O pin. The two logic inputs come from the ORP (see figure 7). One comes from the ORP, and the other comes from the faster ORP bypass. A pair of multiplexers select which signal will be used, and its polarity.

The OE signal comes from the Output Enable. As with the data path, a multiplexer selects the signal polarity. The Output Enable can be set to a logic high (Enabled) when a straight output pin is desired, or logic low (Disabled) when a straight input pin is needed. The Global Reset ($\overline{\text{RESET}}$) signal is driven by the active low chip reset pin. Each I/O Cell can individually select one of the two clock signals (IOCLK 0 or IOCLK 1). These clock signals are generated by the Clock Distribution Network.

Using the multiplexers, the I/O Cell can be configured as an input, an output, a 3-stated output or a bidirectional I/O. The D-type register can be configured as a level sensitive transparent latch or an edge triggered flip-flop to store the in-coming data. Figure 12 illustrates some of the various I/O cell configurations possible.

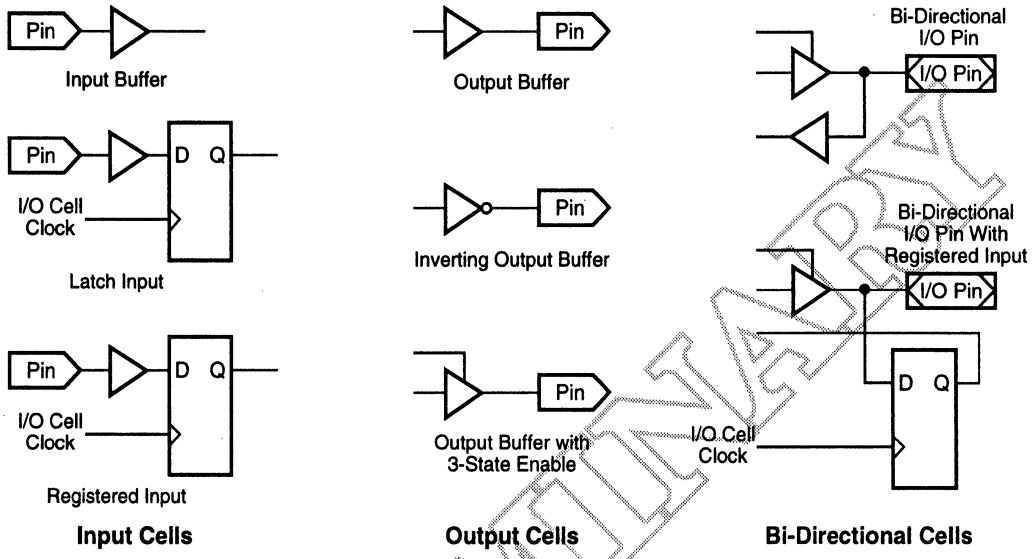
There is an Active Pullup resistor on the I/O pins which is automatically used when the pin is not connected. This prevents the pin from floating and inducing noise into the device or consuming additional power.

Figure 11. I/O Cell Architecture



Architectural Description

Figure 12. Example I/O Cell Configurations



Architectural Description

Clock Distribution Network

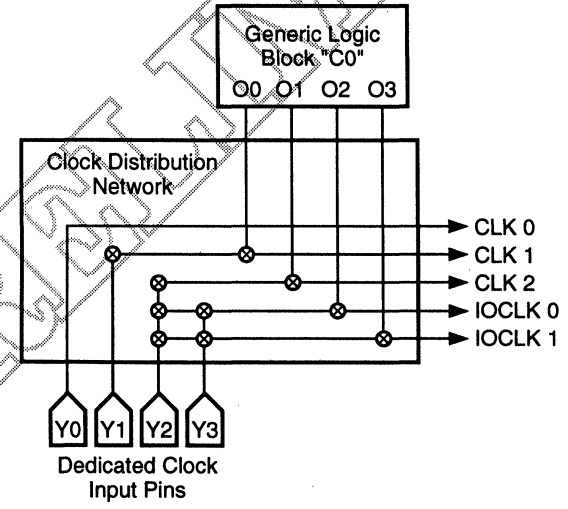
The Clock Distribution Network is shown in figure 13. It generates five global clock signals CLK 0, CLK 1, CLK 2 and IOCLK 0, IOCLK 1. The first three, CLK 0, CLK 1 and CLK 2 are used for clocking all the GLBs in the device. Similarly, IOCLK 0 and IOCLK 1 signals are used for clocking all of the I/O Cells in the device. There are four dedicated system clock pins (Y0, Y1, Y2, Y3) which can be directed to any GLB or any I/O Cell using the Clock Distribution Network. The other inputs to the Clock Distribution Network are the 4 outputs of a dedicated clock GLB ("C0" for ispLSI 1032). These Clock GLB outputs can be used to create a user-defined internal clocking scheme.

Typically the clock GLB will be clocked using an external main clock pin Y0 connected to global clock signal CLK 0. The outputs of the clock GLB in turn will generate "divide by two" or "divide by four" phases of the CLK 0 which can be connected to CLK 1, CLK 2 or IOCLK 0, IOCLK 1 global clocks.

All GLBs also have the capability of generating their own asynchronous clocks using Product Term 12. CLK 0, CLK 1 and CLK 2 feed to their corresponding inputs on all the GLBs (see figure 3).

The two I/O clocks generated in the Clock Distribution Network IOCLK 0 and IOCLK 1, are brought to all 64 of the I/O cells and the user programs the I/O cell to use one of the two. When the Dedicated Clock Input pins Y2 or Y3 are used as one of the I/O Clocks, O2 or O3 from the Clock GLB (C0) cannot be used.

Figure 13. Clock Distribution Network



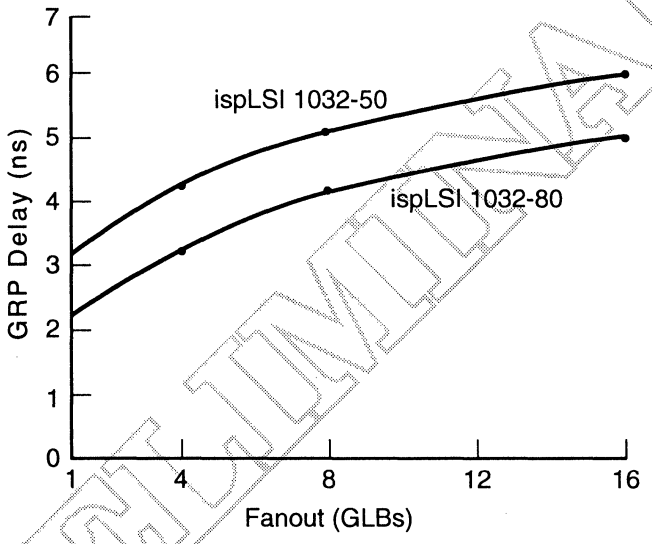
Note: Y3 pin should always be used first as an IOCLK 0 or IOCLK 1 before using Y2 pin.

Architectural Description

Global Routing Pool

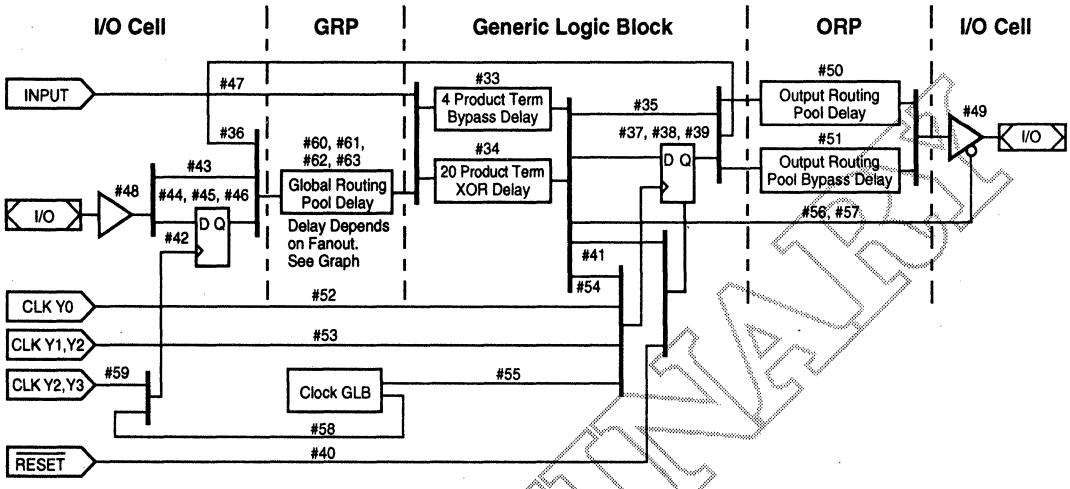
The Global Routing Pool (GRP) is a Lattice proprietary interconnect structure which offers fast predictable speeds with complete connectivity. The GRP allows the outputs from the GLBs or the I/O cell inputs to be connected to the inputs of the GLBs. Any GLB output is available to the input of all other GLBs, and similarly an input from an I/O pin is available as an input to all of the GLBs. Because of the uniform architecture of the ispLSI device, the delays through the Global Routing Pool are both consistent and predictable. However, they are slightly affected by fanout. See the fanout delay graph (Figure 14).

Figure 14. GRP Delay vs Fanout



Timing Model

Figure 15. ispLSI Timing Model



The task of determining the timing through the device is simple and straightforward. The device timing model is shown in figure 15. To determine the time that it takes for data to propagate through the device, simply determine the path the data is expected to follow, and add the various times. Some examples are shown for the critical timing paths used as Data Sheet parameters. Note that the internal timing parameters are given for reference only, and are not tested. (External timing parameters are tested and guaranteed on every device.)

t_{pd1}[1]	=	t_{ib}	+	t_{iobp}	+	t_{grp4}	+	t_{4pt}	+	t_{gbp}	+	t_{mxbp}	+	t_{ob}		
#1	=	#48	+	#43	+	#60	+	#33	+	#35	+	#51	+	#49		
15 ns	=	2	+	0	+	3	+	6	+	0	+	0	+	4		
t_{pd2}[1]	=	t_{ib}	+	t_{iobp}	+	t_{grp4}	+	t_{xor20}	+	t_{gbp}	+	t_{mx}	+	t_{ob}		
#2	=	#48	+	#43	+	#60	+	#34	+	#35	+	#50	+	#49		
20 ns	=	2	+	0	+	3	+	7.5	+	0	+	1	+	4		
f_{max}[1]	=	t_{gco}	+	t_{gfb}	+	t_{grp4}	+	t_{xor20}	+	t_{gsu}						
#11	=	#37	+	#36	+	#60	+	#34	+	#38						
1 / 14ns	=	2	+	0	+	3	+	7.5	+	0						
t_{su2}[1]	=	t_{ib}	+	t_{iobp}	+	t_{grp4}	+	t_{xor20}	-	t_{gy0}	-	t_{gsu}				
#14	=	#48	+	#43	+	#60	+	#34	-	#52	-	#38				
12 ns	=	2	+	0	+	3	+	7.5	-	4	-	0				
t_{co2}[1]	=	t_{gy0}	+	t_{gco}	+	t_{mx}	+	t_{ob}								
#4	=	#52	+	#37	+	#50	+	#49								
13 ns	=	4	+	2	+	1	+	4								

NOTE:
1. The internal delays are rounded and do not necessarily add up to the tested external delays.

Switching Characteristics — ispLSI 1032-80

Internal Timing Model Parameters are not tested and are for reference only

Generic Logic Block (GLB)[1]

PARAMETER	#	DESCRIPTION	MIN.	MAX	UNITS
t_{4pt}	33	4 Product Term Delay	–	6	ns
t_{xor20}	34	20 Product Term Delay	–	7.5	ns
t_{gbp}	35	GLB Register By-Pass Delay	–	0	ns
t_{gfb}	36	GLB Feedback Delay	–	0	ns
t_{gco}	37	GLB Clock to Output Delay	–	2	ns
t_{gsu}	38	GLB Setup Time before Clock	0	–	ns
t_{gh}	39	GLB Hold Time after Clock	2	–	ns

Reset Delays

PARAMETER	#	DESCRIPTION	MIN.	MAX.	UNITS
t_{ggr}	40	GLB Global Reset Delay	–	12	ns
t_{gar}	41	GLB Asynchronous Reset Delay	–	9	ns

Input/Output Cell (I/O Cell)

PARAMETER	#	DESCRIPTION	MIN.	MAX.	UNITS
t_{iat}	42	I/O Cell Latch Delay	–	1	ns
t_{iobp}	43	I/O Cell Register Latch By-Pass Delay	–	0	ns
t_{iosu}	44	I/O Cell Setup Time before Clock/LE	0	–	ns
t_{ioh}	45	I/O Cell Hold Time after Clock/LE	1	–	ns
t_{ioco}	46	I/O Cell Clock/LE to Output Delay	–	1	ns

Input and Output Delays

PARAMETER	#	DESCRIPTION	MIN.	MAX	UNITS
t_{din}	47	Dedicated Input Buffer Delay	–	6	ns
t_{ib}	48	Input Delay for I/O Buffer	–	2	ns
t_{ob}	49	Output Buffer Delay	–	4	ns
t_{mx}	50	Output ORP Delay	–	1	ns
t_{mxbp}	51	Output ORP Bypass Delay	–	0	ns

(Note: Parameter Numbers refer to the timing paths used in the Timing Model Diagram.)

Switching Characteristics — ispLSI 1032-80

Internal AC Characteristics and Conditions are not tested and are for reference only

Clock and Enable Delays, GLB and I/O Cell

PARAMETER	#	DESCRIPTION	MIN.	MAX	UNITS
tgy0	52	Clock Delay, Y0 to GLB	–	4	ns
tgy1/2	53	Clock Delay, Y1 or Y2 to GLB	–	4	ns
tgpt	54	Clock Delay, PT Clk to GLB	–	5	ns
tgcp	55	Clock Delay, Clk GLB to GLB	–	4	ns
tgen	56	Enable Delay, GLB to I/O Cell	–	7	ns
tgdis	57	Disable Delay, GLB to I/O Cell	–	7	ns
tiocp	58	Clock Delay, Clk GLB to I/O Cell	–	4	ns
tioy2/3	59	Clock Delay, Y2 or Y3 to I/O Cell	–	4	ns

GRP Delays

PARAMETER	#	DESCRIPTION	MIN.	MAX	UNITS
tgrp4	60	GRP Delay, Fanout 4	–	3	ns
tgrp8	61	GRP Delay, Fanout 8	–	4	ns
tgrp16	62	GRP Delay, Fanout 16	–	5	ns
tgrp32	63	GRP Delay, Fanout 32	–	8	ns

(Note: Parameter Numbers refer to the timing paths used in the Timing Model Diagram.)

Switching Characteristics — ispLSI 1032-50

Internal Timing Model Parameters are not tested and are for reference only

Generic Logic Block (GLB)[1]

PARAMETER	#	DESCRIPTION	MIN.	MAX	UNITS
t_{4pt}	33	4 Product Term Delay	–	7	ns
t_{xor20}	34	20 Product Term Delay	–	10	ns
t_{gbp}	35	GLB Register By-Pass Delay	–	0	ns
t_{gfb}	36	GLB Feedback Delay	–	0	ns
t_{gco}	37	GLB Clock to Output Delay	–	3	ns
t_{gsu}	38	GLB Setup Time before Clock	2	–	ns
t_{gh}	39	GLB Hold Time after Clock	4	–	ns

Reset Delays

PARAMETER	#	DESCRIPTION	MIN.	MAX.	UNITS
t_{ggr}	40	GLB Global Reset Delay	–	14	ns
t_{gar}	41	GLB Asynchronous Reset Delay	–	10	ns

Input/Output Cell (I/O Cell)

PARAMETER	#	DESCRIPTION	MIN.	MAX.	UNITS
t_{lat}	42	I/O Cell Latch Delay	–	2	ns
t_{iobp}	43	I/O Cell Register Latch By-Pass Delay	–	0	ns
t_{iosu}	44	I/O Cell Setup Time before Clock/LE	0	–	ns
t_{ioh}	45	I/O Cell Hold Time after Clock/LE	2	–	ns
t_{ioco}	46	I/O Cell Clock/LE to Output Delay	–	2	ns

Input and Output Delays

PARAMETER	#	DESCRIPTION	MIN.	MAX	UNITS
t_{din}	47	Dedicated Input Buffer Delay	–	7	ns
t_{ib}	48	Input Delay for I/O Buffer	–	3	ns
t_{ob}	49	Output Buffer Delay	–	5	ns
t_{mx}	50	Output ORP Delay	–	2	ns
t_{mxbp}	51	Output ORP Bypass Delay	–	0	ns

(Note: Parameter Numbers refer to the timing paths used in the Timing Model Diagram.)

Switching Characteristics — ispLSI 1032-50

Internal AC Characteristics and Conditions are not tested and are for reference only

Clock and Enable Delays, GLB and I/O Cell

PARAMETER	#	DESCRIPTION	MIN.	MAX	UNITS
t_{gy0}	52	Clock Delay, Y0 to GLB	–	6	ns
$t_{gy1/2}$	53	Clock Delay, Y1 or Y2 to GLB	–	6	ns
t_{gpt}	54	Clock Delay, PT Clk to GLB	–	7	ns
t_{gcp}	55	Clock Delay, Clk GLB to GLB	–	5	ns
t_{gen}	56	Enable Delay, GLB to I/O Cell	–	9	ns
t_{gdis}	57	Disable Delay, GLB to I/O Cell	–	9	ns
t_{iocp}	58	Clock Delay, Clk GLB to I/O Cell	–	5	ns
$t_{ioy2/3}$	59	Clock Delay, Y2 or Y3 to I/O Cell	–	5	ns

GRP Delays

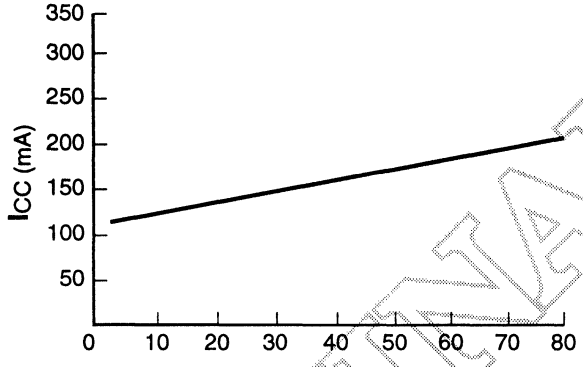
PARAMETER	#	DESCRIPTION	MIN.	MAX	UNITS
t_{grp4}	60	GRP Delay, Fanout 4	–	4	ns
t_{grp8}	61	GRP Delay, Fanout 8	–	6	ns
t_{grp16}	62	GRP Delay, Fanout 16	–	7	ns
t_{grp32}	63	GRP Delay, Fanout 32	–	10	ns

(Note: Parameter Numbers refer to the timing paths used in the Timing Model Diagram.)

Power Consumption

Power Consumption in the ispLSI 1032 device depends on two primary factors: the speed at which the device is operating and the number of Product Terms used. Figure 16 shows the relationship between power and operating speed.

Figure 16. Typical Device Power Consumption vs fmax



Configuration of Eight 16-bit Counters

Security Cell

A security cell is provided in the ispLSI devices to prevent unauthorized copying of the array patterns. Once programmed, this cell prevents further read access to the functional bits in the device. This cell can only be erased by reprogramming the device, so the original configuration can never be examined once this cell is programmed.

Device Programming

ispLSI devices are programmed using a Lattice-approved Device Programmer, available from a number of third party manufacturers or in-system using Lattice programming algorithms. Complete programming of the device takes only a few seconds. Erasing of the device is automatic and is completely transparent to the user.

Latch-up Protection

ispLSI devices are designed with an on-board charge pump to negatively bias the substrate. The negative bias is of sufficient magnitude to prevent input undershoots from causing the internal circuitry to latchup. Additionally, outputs are designed with n-channel pull-ups instead of the traditional p-channel pull-ups to eliminate any possibility of SCR induced latching.

In-System Programmability

The ispLSI devices are the in-system programmable version of the Lattice high density programmable Large Scale Integration (pLSI) devices. By integrating all the high voltage programming circuitry on-chip, the programming can be accomplished by simply shifting data into the device. Once the function is programmed, the non-volatile E^2CMOS cells will not lose the pattern even when the power is turned off.

All necessary programming is done via five TTL level logic interface signals. These five signals are fed into the on-chip programming circuitry where a state machine controls the programming. The simple signals for interface include isp Enable ($\overline{ispEN}$), Serial Data In (SDI), Serial Data Out (SDO), Serial Clock (SCLK) and Mode (MODE) control. Figure 17 illustrates the block diagram of one possible scheme of the programming interface for the isp devices. For details on the operation of the internal state machine and programming of the device please refer to the In-system programming application note.

Figure 17. isp Programming Interface

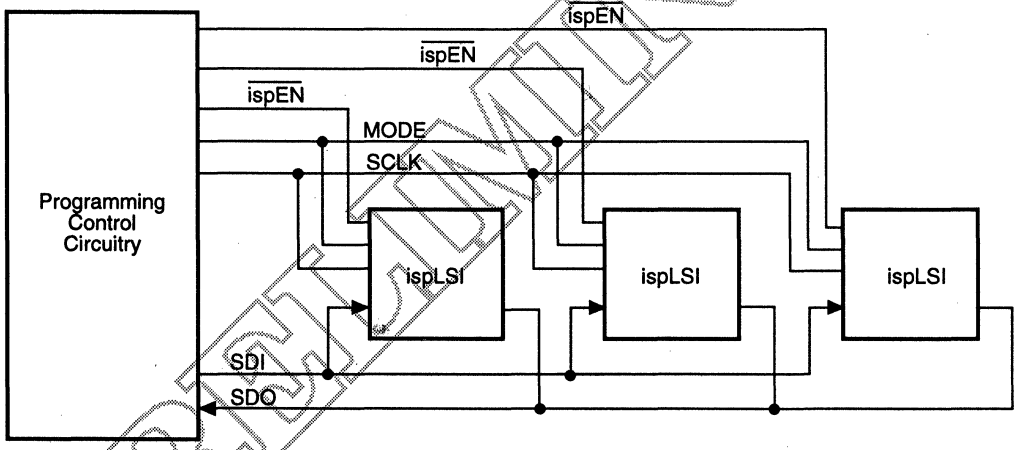
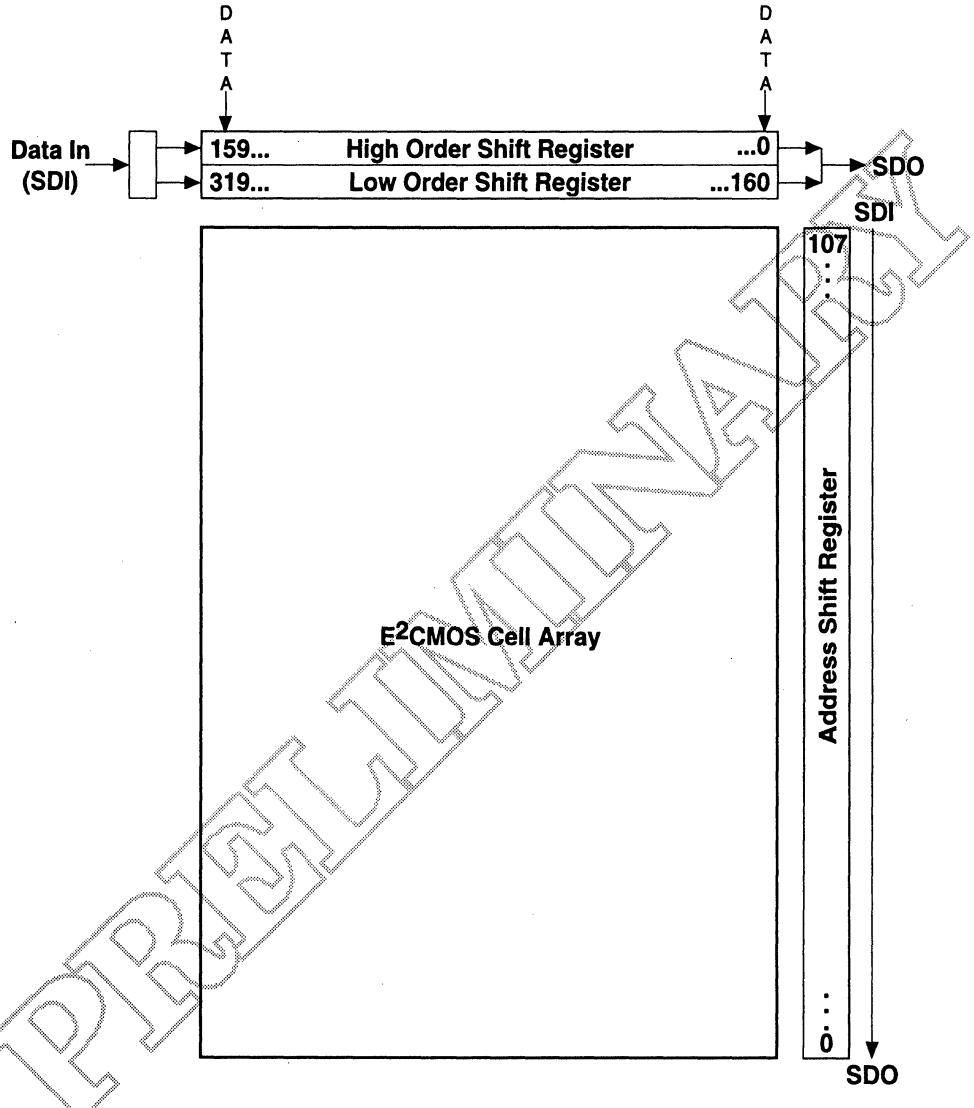


Figure 18. ispLSI Device & Shift Register Layout



Note: A logic "1" in the address shift register enables the row for programming or verification. A logic "0" disables it.

Programming Voltage/Timing Specifications

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNITS
V_{CCP}	Programming Voltage		4.75	5	5.25	V
I_{CCP}	Programming Supply Current ispEN		–	50	100	mA
V_{IHP}	Input Voltage High		2.0	–	V_{CCP}	V
V_{ILP}	Input Vopltage Low		0	–	0.8	V
I_{IP}	Input Current		–	100	200	µA
V_{OHP}	Output Volatge High	I_{OH} = -3.2 mA	2.4	–	V_{CCP}	V
V_{OLP}	Output Voltage Low	I_{OL} =5 mA	0	–	0.5	V
td	Pulse Sequence Delay		1	5	10	µs
tisp	$\overline{\text{ispEN}}$ to Output 3-State		–	1	10	µs
tsu	Setup Time		.1	.5	–	µs
th	Hold Time		.1	.5	–	µs
tclk	Clock Pulse Width		0.5	1	–	µs
tpwv	Verify Pulse Width		20	30	–	µs
tpwp	Programming Pulse Width		40	–	100	ms
tbew	Bulk Erase Pulse Width		200	–	–	ms
trst	Reset Time From Valid V_{CCP}		45	–	–	µs

Figure 19. Timing Waveform for isp Operation

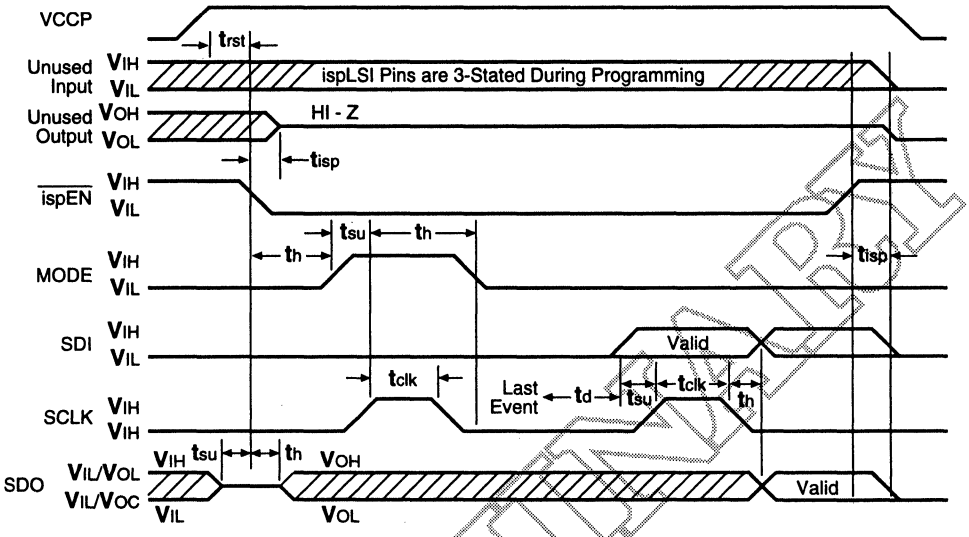
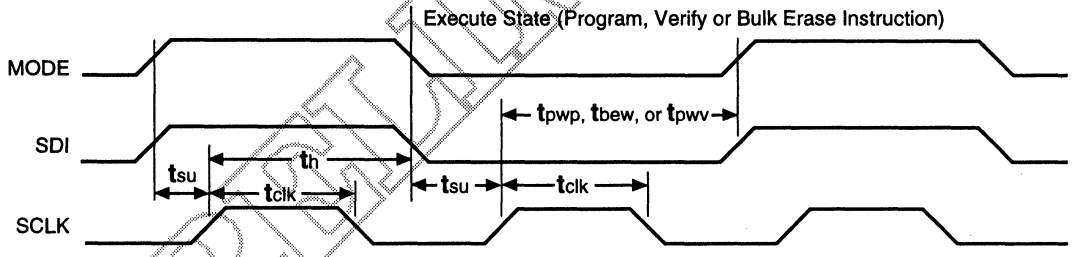
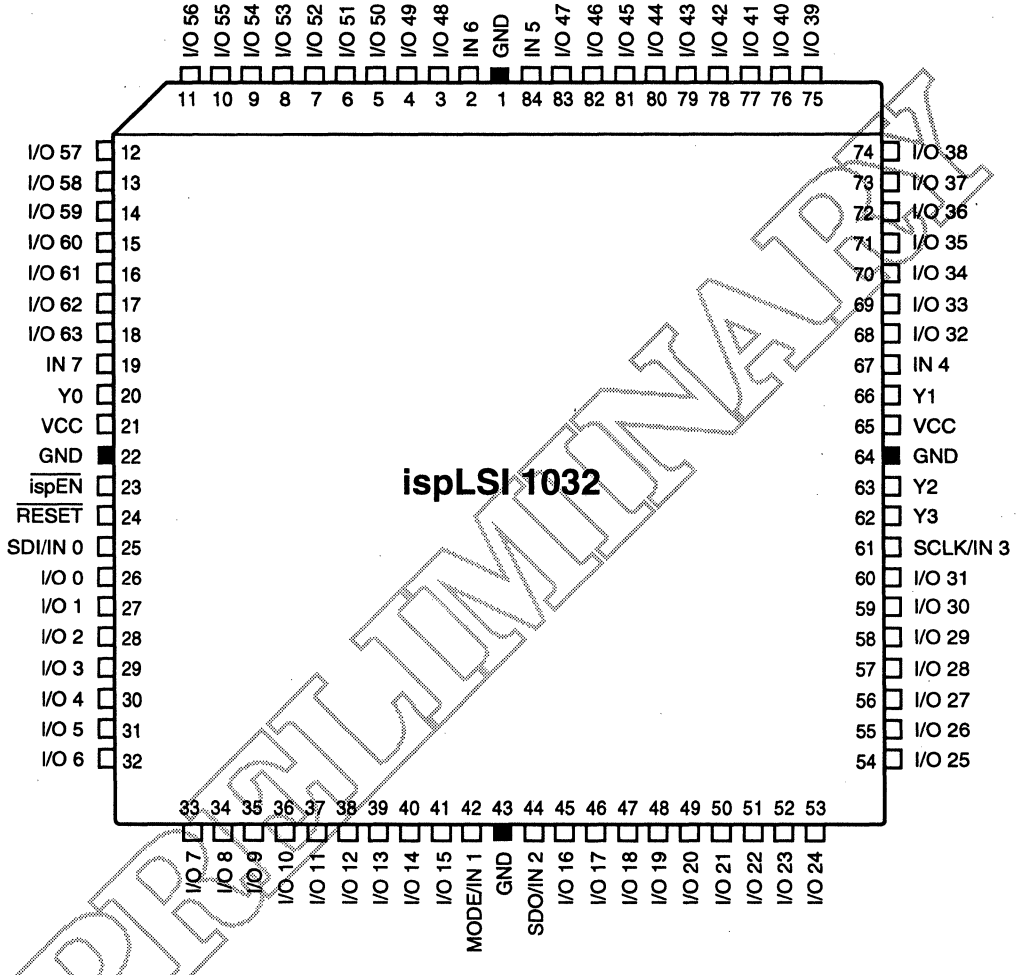


Figure 20. Program, Verify & Bulk Erase Waveform

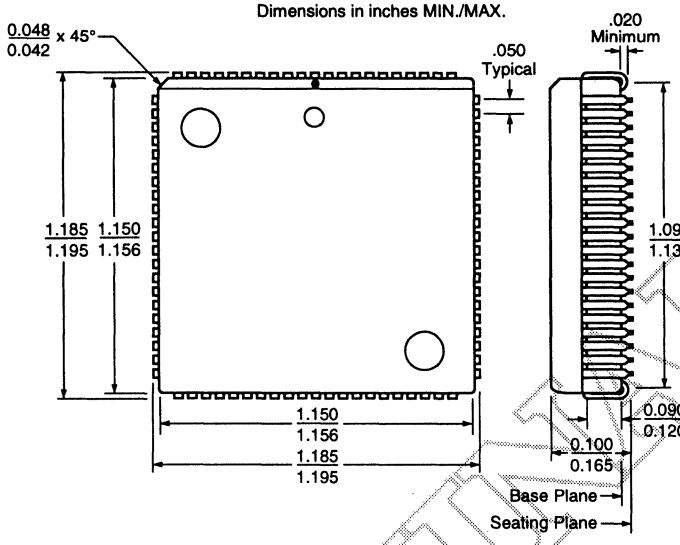


Pin Configuration

ispLSI 1032 PLCC Pinout Diagram



84-Pin PLCC



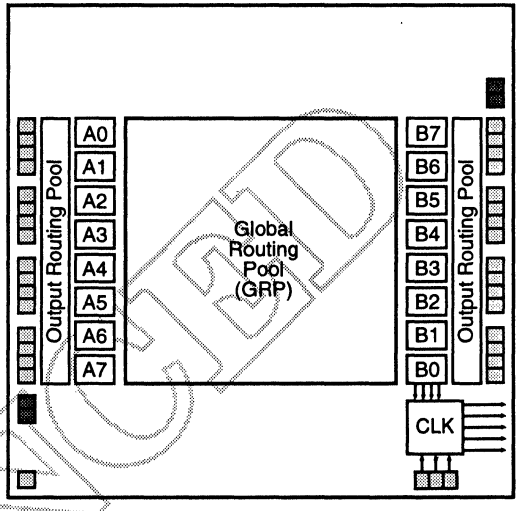
ispLSI™ 1016

in-system programmable Large Scale Integration

Features

- **in-system programmable HIGH DENSITY LOGIC**
  - **— Member of Lattice's ispLSI Family**
  - **— Fully Compatible with Lattice's pLSI™ Family**
  - **— High Speed Global Interconnects**
  - **— 32 I/O Pins, Four Dedicated Inputs**
  - **— 96 Registers**
  - **— Wide Input Gating for Fast Counters, State Machines, Address Decoders, etc.**
  - **— Small Logic Block Size for Random Logic**
  - **— Security Cell Prevents Unauthorized Copying**
- **HIGH PERFORMANCE E²CMOS® TECHNOLOGY**
  - **— f_{max} = 80 MHz Maximum Operating Frequency**
  - **— t_{pd} = 15 ns Propagation Delay**
  - **— TTL Compatible Inputs and Outputs**
- **in-system programmable 5-VOLT ONLY**
  - **— Change Logic and Interconnects "on-the-fly" in Seconds**
  - **— Reprogram Soldered Device for Debugging**
  - **— Non-Volatile E²CMOS Technology**
  - **— 100% Tested**
- **COMBINES EASE OF USE AND THE FAST SYSTEM SPEED OF PLDs WITH THE DENSITY AND FLEXIBILITY OF FIELD PROGRAMMABLE GATE ARRAYS**
  - **— Complete Programmable Device can Combine Glue Logic and Structured Designs**
  - **— 100% Routable at 80% Utilization**
  - **— Three Dedicated Clock Input Pins**
  - **— Synchronous and Asynchronous Clocks**
  - **— Flexible Pin Placement**
  - **— Optimized Global Routing Pool Allows Global Interconnectivity**
- **pLSI/ispLSI™ DEVELOPMENT SYSTEM (pDS™)**
  - **— Boolean Logic Compiler**
  - **— Automatic Place and Route**
  - **— Manual Partitioning**
  - **— PC Platform**
  - **— Easy to Use Windows Interface**
- **ADVANCED pLSI/ispLSI DEVELOPMENT SYSTEM**
  - **— Industry Standard, Third Party Design Environments**
  - **— Schematic Capture**
  - **— Fully Automatic Partitioning**
  - **— Automatic Place and Route**
  - **— Comprehensive Logic and Timing Simulation**
  - **— PC and Workstation Platforms**

Functional Block Diagram



Description

The Lattice ispLSI 1016 is a High Density Programmable Logic Device featuring 5-Volt in-system programmability and in-system diagnostic capabilities. The device contains 96 Registers, 32 Universal I/O pins, four Dedicated Input Pins, three Dedicated Clock Input Pins and a Global Routing Pool (GRP). The GRP provides complete interconnectivity between all of these elements. It is the first device which offers non-volatile "on-the-fly" reprogrammability of the logic, as well as the interconnects to provide truly reconfigurable systems. It is architecturally and parametrically compatible to the pLSI 1016 device, but multiplexes four of the dedicated input pins to control in-system programming.

The basic unit of logic on the ispLSI 1016 device is the Generic Logic Block (GLB). The GLBs are labeled A0, A1 .. B7 (see figure 1). There are a total of 16 GLBs in the ispLSI 1016 device. Each GLB has 18 inputs, a programmable AND/OR/Exclusive OR array, and four outputs which can be configured to be either combinatorial or registered. Inputs to the GLB come from the GRP. All of the GLB outputs are brought back into the GRP so that they can be connected to the inputs of any other GLB on the device.

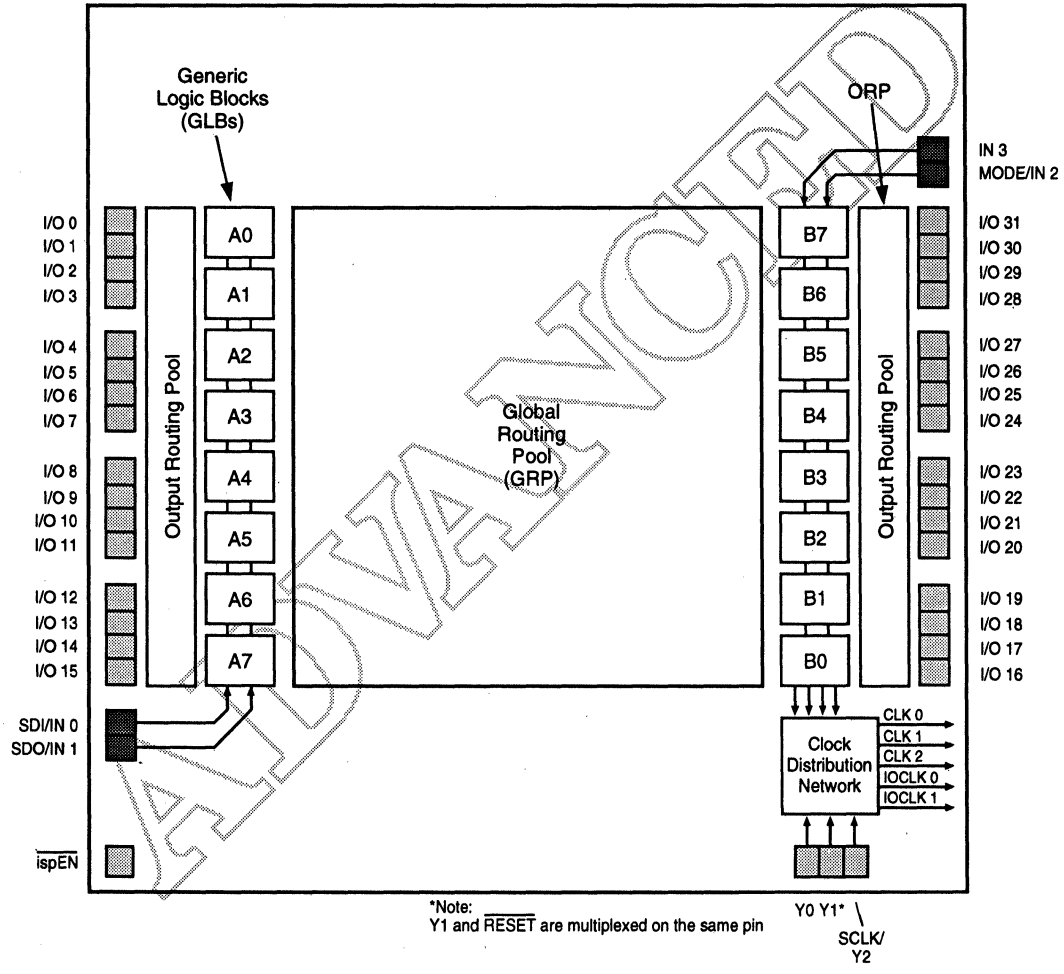
LATTICE SEMICONDUCTOR CORP., 5555 Northeast Moore Ct., Hillsboro, Oregon 97124, U.S.A.
Tel. 1-800-LATTICE (528-8423); FAX (503) 681-3037

January 1992. Rev. A

The device also has 32 I/O Cells, each of which is directly connected to an I/O pin. Each I/O cell can be individually programmed to be a combinatorial input, registered input, latched input, output or bi-directional I/O pin with 3-state. The signal levels are TTL compatible voltages and the output drivers can source 4 mA or sink 8 mA.

Functional Block Diagram

Figure 1. ispLSI 1016



Description (continued)

The 32 I/O Cells are grouped into four sets of 16 each. as shown in figure 1. Each of these I/O groups is associated with a logic Megablock through the use of the Output Routing Pool (ORP) and shares a common Output Enable (OE) signal.

Eight GLBs, 16 I/O Cells and one ORP are connected together to make a logic Megablock. The Megablock is defined by the resources that it shares. The outputs of the eight GLBs are connected to a set of 16 universal I/O cells by the ORP. The ispLSI 1016 Device contains two of these Megablocks.

The GRP has as its inputs the outputs from all of the GLBs and all of the inputs from the bi-directional I/O cells. All of these signals are made available to the inputs of the GLBs. Delays through the GRP have been equalized to minimize timing skew and logic glitching.

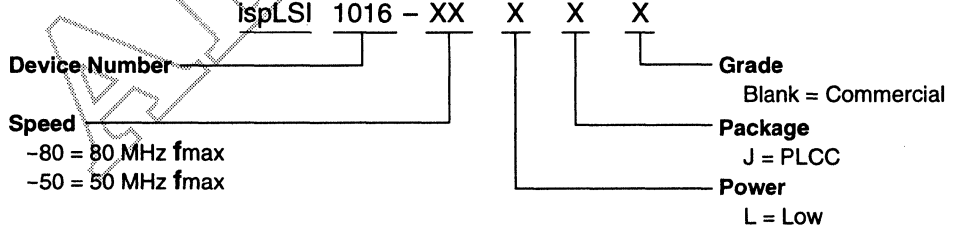
Clocks in the ispLSI 1016 device are selected using the Clock Distribution Network. Three dedicated clock pins (Y0 to Y2) are brought into the distribution network, and five outputs (CLK 0 to CLK 2 and IOCLK 0, IOCLK 1) are provided to route clocks to the GLBs and I/O cells. The Clock Distribution Network can also be driven from a special GLB (B0 on the ispLSI 1016 device). The logic of this GLB allows the user to create an internal clock from a combination of internal signals within the device.

The ispLSI 1016 device is part of Lattice's in-system programmable Large Scale Integration (ispLSI) family. This family contains a range of devices from the ispLSI 1016, with 96 registers, to the ispLSI 1048 with 288 registers. The ispLSI Family Product Selector Guide below lists key attributes of the devices along with the number of resources available.

ispLSI Family Product Selector Guide

DEVICE	ispLSI 1016	ispLSI 1024	ispLSI 1032	ispLSI 1048
GLBs	16	24	32	48
Registers	96	144	192	288
I/O Pins	32	48	64	96
Dedicated Inputs	4	6	8	10
Pin Count	44	68	84	120

Ordering Information



Pin Description

Name	PLCC Pin Numbers	Description
I/O 0 - I/O 3 I/O 4 - I/O 7 I/O 8 - I/O 11 I/O 12 - I/O 15 I/O 16 - I/O 19 I/O 20 - I/O 23 I/O 24 - I/O 27 I/O 28 - I/O 31	15, 16, 17, 18, 19, 20, 21, 22, 25, 26, 27, 28, 29, 30, 31, 32, 37, 38, 39, 40, 41, 42, 43, 44, 3, 4, 5, 6, 7, 8, 9, 10	Input/Output Pins - These are the general purpose I/O pins used by the logic array.
IN 3	2	Dedicated input pins to the device.
$\overline{\text{ispEN}}$	13	Input – Dedicated in-system programming enable input pin. This pin is brought low to enable the programming mode. The MODE, SDI, SDO and SCLK options become active.
SDI/IN 0	14	Input – This pin performs two functions. It is a dedicated input pin when $\overline{\text{ispEN}}$ is logic high. When $\overline{\text{ispEN}}$ is logic low, it functions as an input pin to load programming data into the device. SDI/IN 0 also is used as one of the two control pins for the isp state machine.
MODE/IN 2	36	Input – This pin performs two functions. It is a dedicated input pin when $\overline{\text{ispEN}}$ is logic high. When $\overline{\text{ispEN}}$ is logic low, it functions as a pin to control the operation of the isp state machine.
SDO/IN 1	24	Input/Output – This pin performs two functions. It is a dedicated clock input pin when $\overline{\text{ispEN}}$ is logic high. When $\overline{\text{ispEN}}$ is logic low, it functions as an output pin to read serial shift register data.
SCLK/Y2	33	Input – This pin performs two functions. It is a dedicated clock input when $\overline{\text{ispEN}}$ is logic high. This clock input is brought into the Clock Distribution Network, and can optionally be routed to any GLB and/or I/O Cell on the device. When $\overline{\text{ispEN}}$ is logic low, it functions as a clock pin for the Serial Shift Register.
Y0	11	Dedicated Clock input. This clock input is connected to one of the clock inputs of all of the GLBs on the device.
Y1/$\overline{\text{RESET}}$	35	This pin performs two functions: – Dedicated clock input. This clock input is brought into the Clock Distribution Network, and can optionally be routed to any GLB and/or I/O Cell on the device. – Active Low (0) Reset pin which resets all of the GLB and I/O registers in the device.
GND	1, 23	Ground (GND)
VCC	12, 34	V_{CC}

Absolute Maximum Ratings [1]

Supply Voltage V_{CC} . . . . . -0.5 to +7.0V

Input Voltage Applied. . . . . .-2.5 to V_{CC} +1.0V

Off-State Output Voltage Applied . . . . . .-2.5 to V_{CC} +1.0V

Storage Temperature . . . . . .-65 to 125°C

Ambient Temp. with Power Applied . . . . . .-55 to 125°C

1. Stresses above those listed under the "Absolute Maximum Ratings" may cause permanent damage to the device. Functional operation of the device at these or at any other conditions above those indicated in the operational sections of this specification is not implied (while programming, follow the programming specifications).

DC Recommended Operating Condition

SYMBOL	PARAMETER	MIN.	MAX.	UNITS
T_A	Ambient Temperature	0	70	°C
V_{CC}	Supply Voltage	4.75	5.25	V
V_{IL}	Input Low Voltage	0	0.8	V
V_{IH}	Input High Voltage	2.0	V_{CC}	V

Capacitance (T_A=25°C,f=1.0 MHz)

SYMBOL	PARAMETER	MAXIMUM[1]	UNITS	TEST CONDITIONS
C_1	Input Capacitance	8	pf	V_{CC}=5.0V, V_{IN}=2.0V
C_2	I/O, Y Capacitance	10	pf	V_{CC}=5.0V, VI/O, Y=2.0V

1. Guaranteed but not 100% tested.

Data Retention Specifications

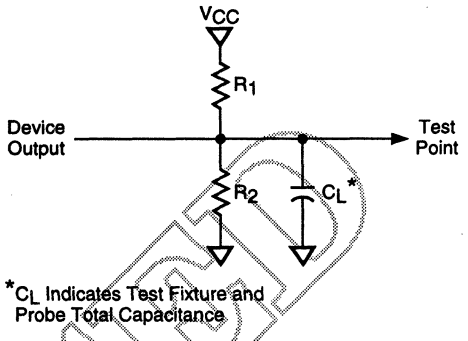
PARAMETER	MINIMUM	MAXIMUM	UNITS
Data Retention	20	–	YEARS
Erase/Reprogram Cycles	–	1000	CYCLES

Switching Test Conditions

Input Pulse Levels	GND to 3.0V
Input Rise and Fall Times	3ns 10% to 90%
Input Timing Reference Levels	1.5V
Output Timing Reference Levels	1.5V
Output Load	See Figure 2

3-state levels are measured 0.5V from steady-state active level.

Figure 2. Test Load



Output Load Conditions (see figure 2)

Test Condition		R1	R2	CL
1		470Ω	390Ω	35pF
2	Active High	∞	390Ω	35pF
	Active Low	470Ω	390Ω	35pF
3	Active High to Z at V_{OH} - 0.5V	∞	390Ω	5pF
	Active Low to Z at V_{OL} + 0.5V	470Ω	390Ω	5pF

DC Electrical Characteristics

Over Recommended Operating Conditions

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNITS
V_{OL}	Output Low Voltage	I_{OL} =8 mA.	–	–	0.4	V
V_{OH}	Output High Voltage	I_{OH} =-4 mA.	2.4	–	–	V
I_{IL}	Input or I/O Low Leakage Current	0V ≤ V_{IN} ≤ V_{IL} (MAX.)	–	–	-10	µA
I_{IH}	Input or I/O High Leakage Current	V_{IH} ≤ V_{IN} ≤ V_{CC}	–	–	10	µA
I_{OS}[1]	Output Short Circuit Current	V_{CC} = 5V, V_{OUT} = 0.5V	-60	–	-200	mA
I_{CC}[2]	Operating Power Supply Current	V_{IL} = 0.5V, V_{IH} = 3.0V F_{TOGGLE} = 20 MHz	–	–	–	mA

1. One output at a time for a maximum duration of one second. (Vout = 0.5V)
2. Measured at a frequency of 20 MHz using four 16-bit counters.

External Switching Characteristics[1, 2, 3] — ispLSI 1016-80

Over Recommended Operating Conditions

PARAMETER	TEST[6] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
t_{pd1}	1	1	Data Propagation Delay, 4PT bypass, ORP bypass	–	12	15	ns
t_{pd2}	1	2	Data Propagation Delay	–	15	20	ns
t_{co1}[5]	1	3	External Clock to Output Delay, ORP bypass	–	8	11	ns
t_{co2}[5]	1	4	External Clock to Output Delay	–	9	14	ns
t_{co3}	1	5	Internal Synch. Clock to Output Delay	–	15	20	ns
t_{co4}	1	6	Asynchronous Clock to Output Delay	–	13	20	ns
t_{r1}	1	7	External Pin Reset to Output Delay	–	13	20	ns
t_{r2}	1	8	Asynchronous PT Reset to Output Delay	–	15	22	ns
t_{en}	2	9	Input to Output Enable	–	13	20	ns
t_{dis}	3	10	Input to Output Disable	–	13	20	ns

External AC Recommended Operating Conditions[1, 2, 3] — ispLSI 1016-80

Over Recommended Operating Conditions

PARAMETER	TEST[6] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
f_{max}[4]	1	11	Clock Frequency with Internal Feedback	–	100	80	MHz
f_{max} (External)	1	12	Clock Frequency with External Feedback	–	70	50	MHz
t_{su1}	–	13	Setup Time before External Synch. Clock, 4PT bypass	9	6	–	ns
t_{su2}	–	14	Setup Time before External Synch Clock	12	8	–	ns
t_{su3}	–	15	Setup Time before Internal Synch. Clock	9	3	–	ns
t_{su4}	–	16	Setup Time before Asynchronous Clock	9	4	–	ns
t_{h1}	–	17	Hold time after External Synchronous Clock, 4PT bypass	2	-1	–	ns
t_{h2}	–	18	Hold time after External Synchronous Clock	2	-1	–	ns
t_{h3}	–	19	Hold time after Internal Synchronous Clock	8	2	–	ns
t_{h4}	–	20	Hold time after Asynchronous Clock	8	1	–	ns
t_{rw1}	–	21	External Reset Pulse Duration	10	8	–	ns
t_{rw2}	–	22	Asynchronous Reset Pulse Duration	10	8	–	ns
t_{wh1}, t_{wl1}	–	23,24	External Synchronous Clock Pulse Duration, High, Low	6	5	–	ns
t_{wh2}, t_{wl2}	–	25,26	Asynchronous Clock Pulse Duration, High, Low	6	5	–	ns

1. External Parameters are tested and guaranteed.
2. See Timing Technical Note for further details.
3. Unless noted, all parameters use ORP, GRP fanout of four, PTSA, and are measured with 16 outputs switching.
4. Standard 16-bit counter implementation using GRP feedback.
5. Clock to output specifications include a maximum skew of 2 ns.
6. Refer to Switching Test Conditions section.

Switching Characteristics[1, 2, 3] — ispLSI 1016-80

Using I/O Cell

PARAMETER	TEST[4] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
t_{su5}	–	27	Setup Time before External Synchronous Clock	5	0	–	ns
t_{su6}	–	28	Setup Time before Internal Synchronous Clock	0	-3	–	ns
t_{h5}	–	29	Hold Time after External Synchronous Clock	8	4	–	ns
t_{h6}	–	30	Hold Time after Internal Synchronous Clock	15	11	–	ns
t_{wh3}, t_{wl3}	–	31,32	Clock Pulse Duration, High, Low	6	5	–	ns

1. External Parameters are tested and guaranteed.
2. See Timing Technical Note for further details.
3. Unless noted, all parameters use ORP, GRP fanout of four, PTSA, and are measured with 16 outputs switching.
4. Refer to Switching Test Conditions section.

External Switching Characteristics[1, 2, 3] — ispLSI 1016-50

Over Recommended Operating Conditions

PARAMETER	TEST[6] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
t_{pd1}	1	1	Data Propagation Delay, 4PT bypass, ORP bypass	–	16	20	ns
t_{pd2}	1	2	Data Propagation Delay, ORP	–	19	25	ns
t_{co1}[5]	1	3	External Clock to Output Delay, ORP bypass	–	12	16	ns
t_{co2}[5]	1	4	External Clock to Output Delay	–	15	20	ns
t_{co3}	1	5	Internal Synch. Clock to Output Delay	–	21	28	ns
t_{co4}	1	6	Asynchronous Clock to Output Delay	–	21	28	ns
t_{r1}	–	7	External Pin Reset to Output Delay	–	21	28	ns
t_{r2}	–	8	Asynchronous PT Reset to Output Delay	–	24	30	ns
t_{en}	2	9	Input to Output Enable	–	21	28	ns
t_{dis}	3	10	Input to Output Disable	–	21	28	ns

External AC Recommended Operating Conditions[1, 2, 3] — ispLSI 1016-50

Over Recommended Operating Conditions

PARAMETER	TEST[6] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
f_{max}[4]	1	11	Clock Frequency with Internal Feedback	–	70	50	MHz
f_{max} (External)	1	12	Clock Frequency with External Feedback	–	45	33	MHz
t_{su1}	–	13	Setup Time before External Synch. Clock, 4PT bypass	14	10	–	ns
t_{su2}	–	14	Setup Time before External Synch Clock	17	13	–	ns
t_{su3}	–	15	Setup Time before Internal Synch. Clock	13	9	–	ns
t_{su4}	–	16	Setup Time before Asynchronous Clock	13	9	–	ns
t_{h1}	–	17	Hold time after External Synchronous Clock, 4PT bypass	7	3	–	ns
t_{h2}	–	18	Hold time after External Synchronous Clock	7	3	–	ns
t_{h3}	–	19	Hold time after Internal Synchronous Clock	11	5	–	ns
t_{h4}	–	20	Hold time after Asynchronous Clock	11	5	–	ns
t_{rw1}	–	21	External Reset Pulse Duration	15	13	–	ns
t_{rw2}	–	22	Asynchronous Reset Pulse Duration	15	13	–	ns
t_{wh1}, t_{wl1}	–	23,24	External Synchronous Clock Pulse Duration, High, Low	10	8	–	ns
t_{wh2}, t_{wl2}	–	25,26	Asynchronous Clock Pulse Duration, High, Low	10	8	–	ns

1. External Parameters are tested and guaranteed.
2. See Timing Technical Note for further details.
3. Unless noted, all parameters use ORP, GRP fanout of four, PTSA, and are measured with 16 outputs switching.
4. Standard 16-bit counter implementation using GRP feedback.
5. Clock to output specifications include a maximum skew of 2 ns.
6. Refer to Switching Test Conditions section.

Switching Characteristics[1, 2, 3] — ispLSI 1016-50

Using I/O Cell

PARAMETER	TEST[4] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
t_{su5}	–	27	Setup Time before External Synchronous Clock	10	5	–	ns
t_{su6}	–	28	Setup Time before Internal Synchronous Clock	0	-5	–	ns
t_{h5}	–	29	Hold Time after External Synchronous Clock	12	6	–	ns
t_{h6}	–	30	Hold Time after Internal Synchronous Clock	20	15	–	ns
t_{wh3}, t_{wl3}	–	31,32	Clock Pulse Duration, High, Low	10	8	–	ns

1. External Parameters are tested and guaranteed.
2. See Timing Technical Note for further details.
3. Unless noted, all parameters use ORP, GRP fanout of four, PTSA, and are measured with 16 outputs switching.
4. Refer to Switching Test Conditions section.

Architectural Description

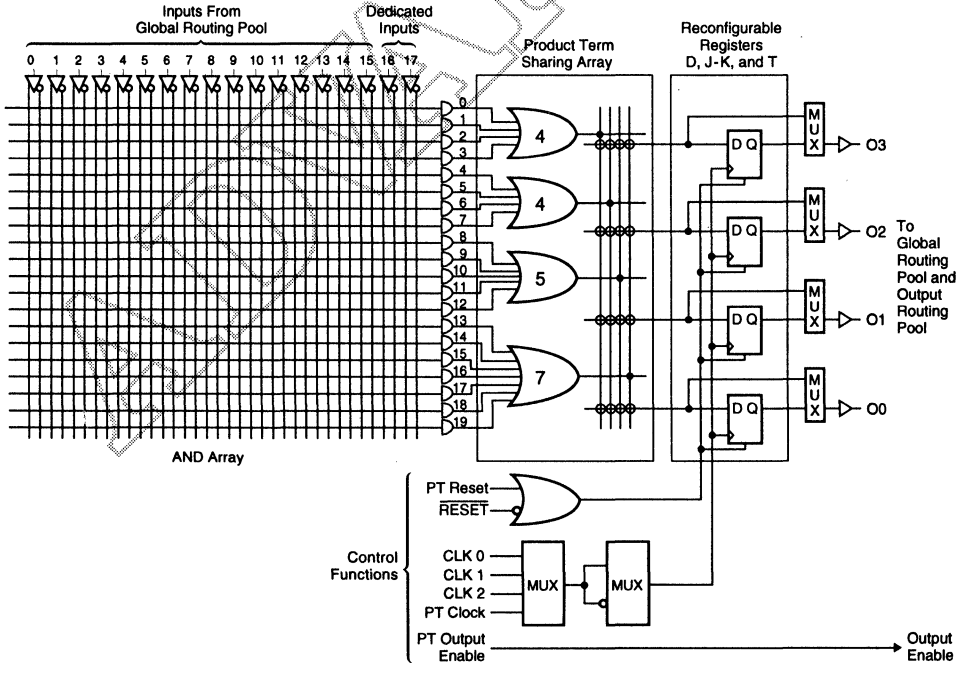
The Generic Logic Block

The Generic Logic Block (GLB) is the standard logic block of the Lattice High Density ispLSI Device. This GLB has 18 inputs, four outputs and the logic necessary to implement most standard logic functions. The internal logic of the GLB is divided into four separate sections (see figure 3). The AND Array, the Product Term Sharing Array (PTSA), the Reconfigurable Registers, and the Control Functions. The AND array consists of 20 product terms which can produce the logical sum of any of the 18 GLB inputs. Sixteen of the inputs come from the Global Routing Pool, and are either feedback signals from any of the 16 GLBs or inputs from the external I/O Cells. The two remaining inputs come directly from two dedicated input pins. These signals are available to the product terms in both the logical true and the complemented forms which makes boolean logic reduction easier.

The PTSA takes the 20 product terms and allocates them to the four GLB outputs. There are four OR gates, with four, four, five and seven product terms (see figure 3). The output of any of these gates can be routed to any of the four GLB outputs, and if more product terms are needed, the PTSA can combine them as necessary. If the users' main concern is speed, the PTSA can use a bypass circuit with four product terms to increase the performance of the cell (see figure 4). This can be done to any or all of the four outputs from the GLB.

The Reconfigurable Registers consist of four D-type flip-flops with an Exclusive OR Gate on the input. The Exclusive OR gate in the GLB can be used either as a logic element or to reconfigure the D-type flip-flop to emulate a J-K, or T-type flip-flop (see figure 5). This greatly simplifies the design of counters, comparators and ALU type functions. The registers can be bypassed if the user needs a combinatorial output. Each register output is brought back into the Global Routing Pool and is also brought to the I/O cells via the Output Routing Pool. Reconfigurable registers are not available when the four product term bypass is used.

Figure 3. GLB: Product Term Sharing Array



Architectural Description

Figure 4. GLB: Four Product Term Bypass

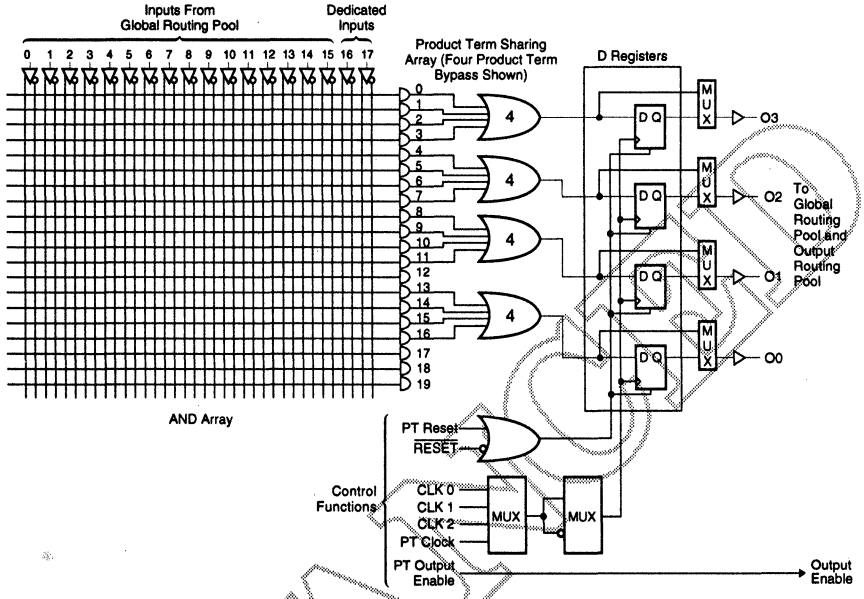
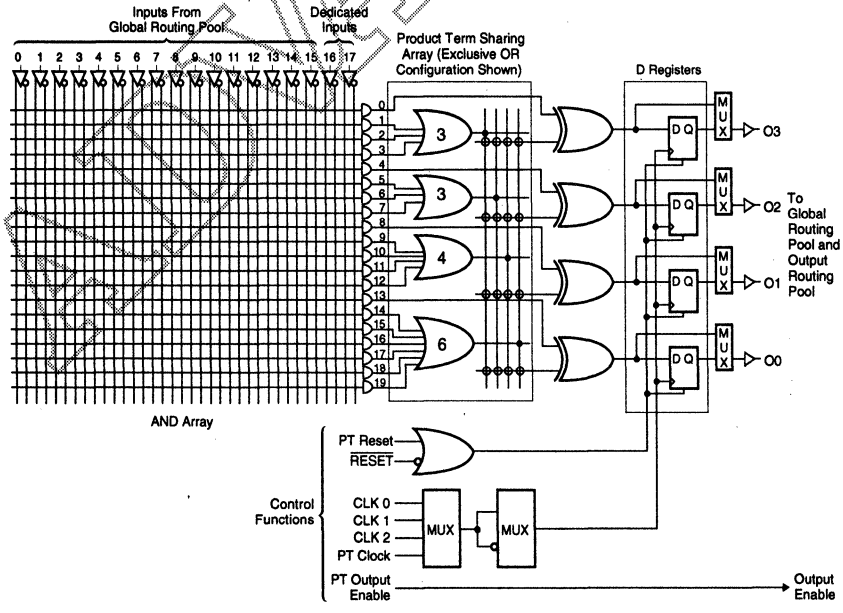


Figure 5. GLB: Exclusive OR Gate



Architectural Description

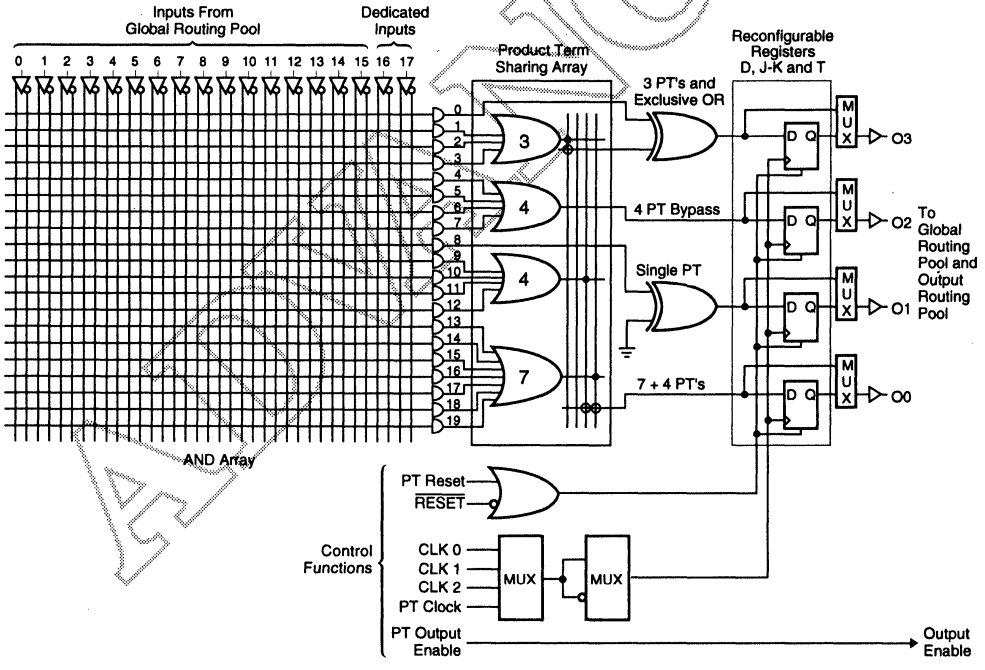
The Generic Logic Block (continued)

The PTSA is flexible enough to allow these features to be used in virtually any combination that the user desires. In the GLB shown in figure 6, Output Three (O3) is configured using the XOR Gate and Output Two (O2) is configured using the four Product Term Bypass. Output One (O1) uses one of the inputs from the five Product Term OR gate while Output Zero (O0) combines the remaining four product terms with all of the product terms from the seven Product Term OR gate for a total of eleven (7+4).

Various signals which control the operation of the GLB are developed in the Control Function section. The clock for the registers can come from any of three sources developed in the Clock Distribution Network (See Clock Distribution Network Section) or from a product term within the GLB. The Reset Signal for the GLB can come from the Global Reset pin or from a product term within the block. The Output Enable for the I/O cells associated with the GLB comes from a product term within the block. Use of a product term for a control function makes that product term unavailable for use as a logic term. Refer to the following table to determine which logic functions are affected.

There are additional features in a GLB which allow implementation of logic intensive functions. These features are accessible using the hard Macros from the software and require no intervention on the part of the user.

Figure 6. GLB: Various Logical Combinations



Architectural Description

Product Term Sharing Matrix

Product Term #	Standard Configuration Output Number 3 2 1 0	Four Product Term Bypass Output Number 3 2 1 0	Single Product Term Output Number 3 2 1 0	XOR Function Output Number 3 3 2 2 1 1 0 0	Alternate Function
0	■ ■ ■ ■	■	■	■	
1	■ ■ ■ ■	■		■	
2	■ ■ ■ ■	■		■	
3	■ ■ ■ ■	■		■	
4	■ ■ ■ ■	■	■	■	
5	■ ■ ■ ■	■		■	
6	■ ■ ■ ■	■		■	
7	■ ■ ■ ■	■		■	
8	■ ■ ■ ■	■	■	■	
9	■ ■ ■ ■	■		■	
10	■ ■ ■ ■	■		■	
11	■ ■ ■ ■	■		■	
12	■ ■ ■ ■	■		■	■ CLK/Reset
13	■ ■ ■ ■	■	■	■	
14	■ ■ ■ ■	■		■	
15	■ ■ ■ ■	■		■	
16	■ ■ ■ ■	■		■	
17	■ ■ ■ ■			■	
18	■ ■ ■ ■			■	
19	■ ■ ■ ■			■	■ OE/Reset

This matrix shows how each of the product terms are used in the various modes. As an example, Product Term 12 can be used as an input to the five input OR gate in the standard configuration. This OR gate under standard configuration can be routed to any of the four GLB outputs. Product Term 12 is not used in the four product term bypass mode. When GLB output one is used in the Exclusive OR (XOR) mode Product Term 12 becomes one of the inputs to the four input OR Gate. If Product Term 12 is not used in the logic, then it is available for use as either the Asynchronous Clock signal or the GLB Reset signal.

Architectural Description

The Megablock

The ispLSI family is structured into multiple "Megablocks". A Megablock consists of 8 GLBs, an Output Routing Pool (ORP) and 16 I/O Cells. Each of these will be explained in detail in the following sections. These elements are coupled together as shown in figure 7. This logic structure is referred to as the Megablock. The various members of the ispLSI family are created by combining from two to six Megablocks on a single device.

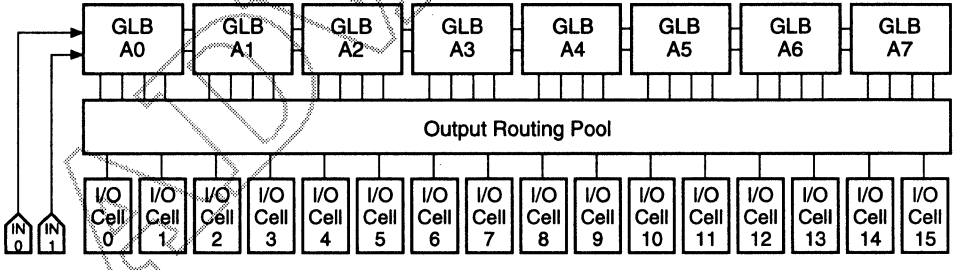
The Megablock shares two sets of resources. The eight GLBs within the Megablock share two dedicated input pins. These dedicated input pins are not available to GLBs in any other Megablock. These pins are dedicated (non-registered) inputs only and are automatically assigned by software. One Output Enable signal is generated within the Megablock and is common to all sixteen of the I/O Cells in the Megablock. The Output Enable signal can be generated using a product term in any of the eight GLBs within the Megablock (See the following section on the Output Enable Multiplexers).

Because of the shared logic within the Megablock, signals which share a common function (counters, busses, etc.) should be grouped within a Megablock. This will allow the user to obtain the best utilization of the logic within the device and eliminate routing bottlenecks.

Megablocks in Each Device

DEVICE	MEGABLOCKS	GLBs	I/O CELLS
ispLSI 1016	2	16	32
ispLSI 1024	3	24	48
ispLSI 1032	4	32	64
ispLSI 1048	6	48	96

Figure 7. The Megablock



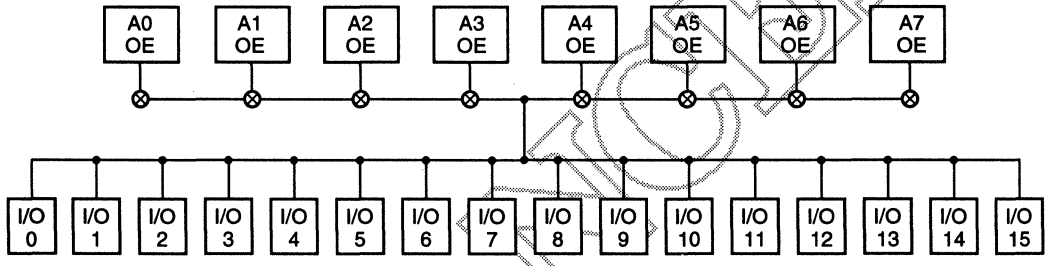
Note: The inputs from the I/O Cell are not shown in this figure, they go directly to the GRP.

Architectural Description

The Output Enable Control

One Output Enable (OE) signal can be generated within each GLB using the OE Product Term 19. One of the eight OE signals from each GLB within a Megablock, is then routed to all of the I/O Cells within the Megablock (see figure 8). This OE signal can simultaneously control all of the 16 I/O cells which are used in 3-state mode. Individual I/O cells also have independent control for permanently enabling or disabling the output buffer (Refer to the I/O Cell Section). Only one OE signal is allowed per Megablock for 3-state operation. The advantage to this approach is that the OE signal can be generated in any GLB within the Megablock which happens to have an unused OE product term. This frees up the other OE product terms for use as logic.

Figure 8. Output Enable Controls



Architectural Description

The Output Routing Pool

The Output Routing Pool (ORP) routes signals from the Generic Logic Block outputs to I/O Cells configured as outputs or bi-directional pins (see figure 9). The purpose of the ORP is to allow greater flexibility when assigning I/O pins. It also simplifies the job for the routing software which results in a higher degree of utilization.

By examining the programmable switch matrix in figure 9, it can be seen that a GLB output can be connected to one of four I/O Cells. Further flexibility is provided by using the PTSA, (Figures 3 through 7) which makes the GLB outputs completely interchangeable. This allows the routing program to freely interchange the outputs to achieve the best routability. This is an automatic process and requires no intervention on the part of the user.

The ORP bypass connections (see figure 10) further increase the flexibility of the device. The ORP bypass connect specific GLB outputs to specific I/O Cells at a faster speed. The bypass path tends to restrict the routability of the device and should only be used for critical signals.

Figure 9. Output Routing Pool

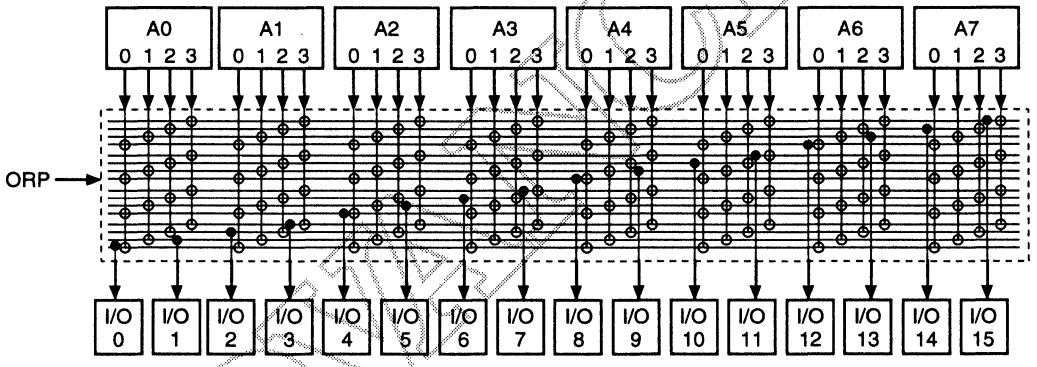
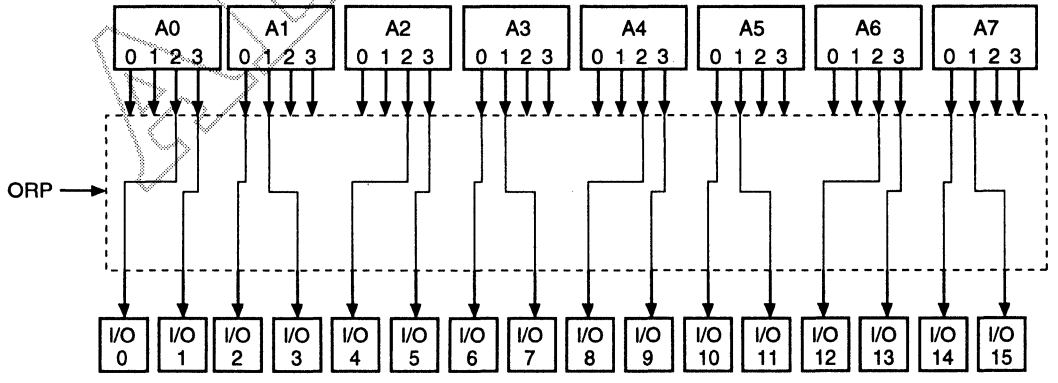


Figure 10. Output Routing Pool showing Bypass



Architectural Description

I/O Cell

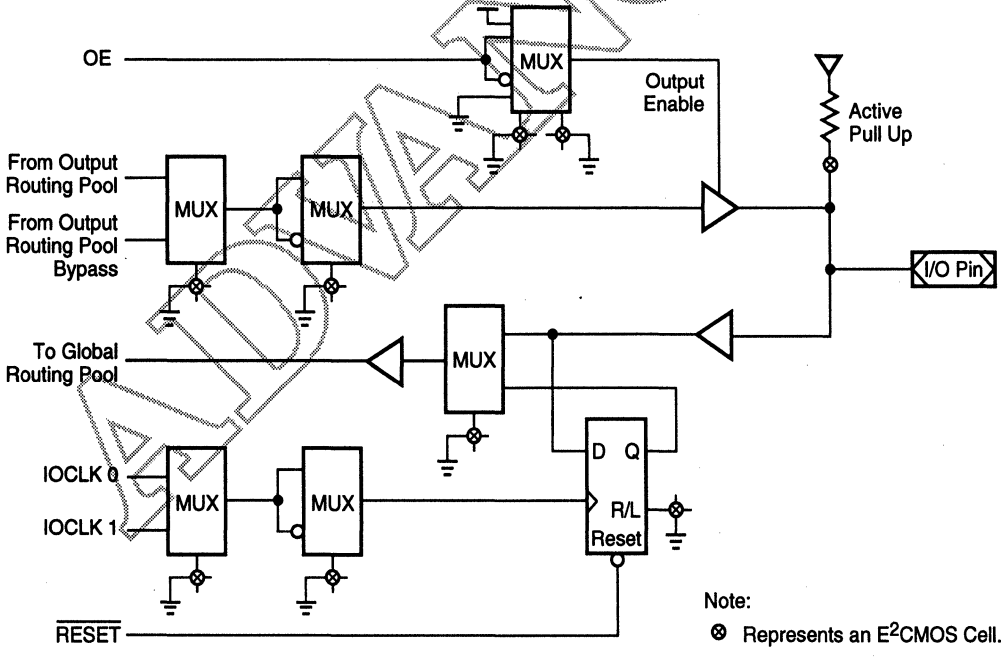
The I/O Cell (see figure 11) is used to route input, output or bidirectional signals connected to the I/O pin. The two logic inputs come from the ORP (see figure 7). One comes from the ORP, and the other comes from the faster ORP bypass. A pair of multiplexers select which signal will be used, and its polarity.

The OE signal comes from the Output Enable. As with the data path, a multiplexer selects the signal polarity. The Output Enable can be set to a logic high (Enabled) when a straight output pin is desired, or logic low (Disabled) when a straight input pin is needed. The Global Reset ($\overline{\text{RESET}}$) signal is driven by the active low chip reset pin. Each I/O Cell can individually select one of the two clock signals (IOCLK 0 or IOCLK 1). These clock signals are generated by the Clock Distribution Network.

Using the multiplexers, the I/O Cell can be configured as an input, an output, a 3-stated output or a bidirectional I/O. The D-type register can be configured as a level sensitive transparent latch or an edge triggered flip-flop to store the in-coming data. Figure 12 illustrates some of the various I/O cell configurations possible.

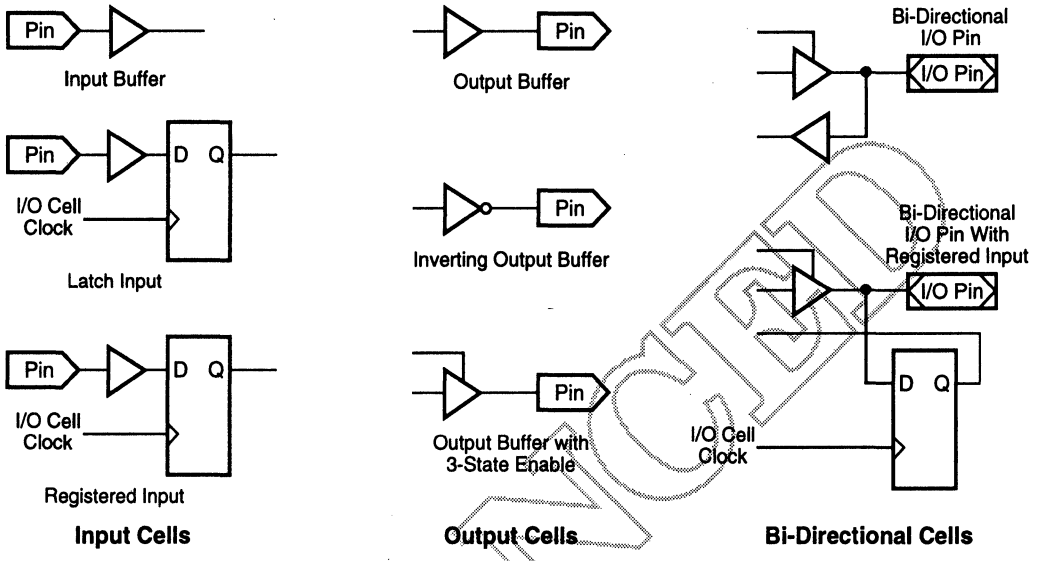
There is an Active Pullup resistor on the I/O pins which is automatically used when the pin is not connected. This prevents the pin from floating and inducing noise into the device or consuming additional power.

Figure 11. I/O Cell Architecture



Architectural Description

Figure 12. Example I/O Cell Configurations



Architectural Description

Clock Distribution Network

The Clock Distribution Network is shown in figure 13. It generates five global clock signals CLK 0, CLK 1, CLK 2 and IOCLK 0, IOCLK 1. The first three, CLK 0, CLK 1 and CLK 2 are used for clocking all the GLBs in the device. Similarly, IOCLK 0 and IOCLK 1 signals are used for clocking all of the I/O Cells in the device. There are three dedicated system clock pins (Y0, Y1, Y2) which can be directed to any GLB or any I/O Cell using the Clock Distribution Network. The other inputs to the Clock Distribution Network are the 4 outputs of a dedicated clock GLB ("B0" for ispLSI 1016). These Clock GLB outputs can be used to create a user-defined internal clocking scheme.

Typically the clock GLB will be clocked using an external main clock pin Y0 connected to global clock signal CLK 0. The outputs of the clock GLB in turn will generate "divide by two" or "divide by four" phases of the CLK 0 which can be connected to CLK 1, CLK 2 or IOCLK 0, IOCLK 1 global clocks.

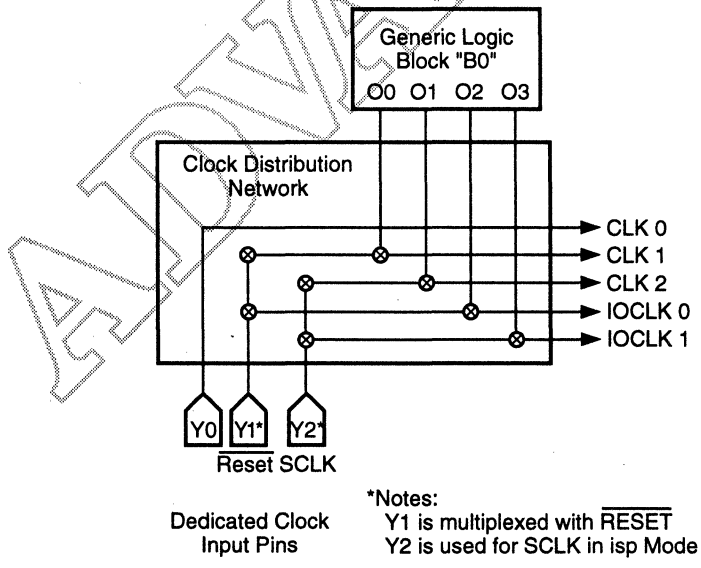
All GLBs also have the capability of generating their own asynchronous clocks using Product Term 12. CLK 0, CLK 1 and CLK 2 feed to their corresponding inputs on all the GLBs (see figure 3).

The two I/O clocks generated in the Clock Distribution Network IOCLK 0 and IOCLK 1, are brought to all 64 of the I/O cells and the user programs the I/O cell to use one of the two.

Global Routing Pool

The Global Routing Pool (GRP) is a Lattice proprietary interconnect structure which offers fast predictable speeds with complete connectivity. The GRP allows the outputs from the GLBs or the I/O cell inputs to be connected to the inputs of the GLBs. Any GLB output is available to the input of all other GLBs, and similarly an input from an I/O pin is available as an input to all of the GLBs. Because of the uniform architecture of the ispLSI device, the delays through the Global Routing Pool are both consistent and predictable. However, they are slightly affected by fanout.

Figure 13. Clock Distribution Network



Security Cell

A security cell is provided in the ispLSI devices to prevent unauthorized copying of the array patterns. Once programmed, this cell prevents further read access to the functional bits in the device. This cell can only be erased by reprogramming the device, so the original configuration can never be examined once this cell is programmed.

Device Programming

ispLSI devices are programmed using a Lattice-approved Device Programmer, available from a number of third party manufacturers or in-system using Lattice programming algorithms. Complete programming of the device takes only a few seconds. Erasing of the device is automatic and is completely transparent to the user.

Latch-up Protection

ispLSI devices are designed with an on-board charge pump to negatively bias the substrate. The negative bias is of sufficient magnitude to prevent input undershoots from causing the internal circuitry to latchup. Additionally, outputs are designed with n-channel pull-ups instead of the traditional p-channel pull-ups to eliminate any possibility of SCR induced latching.

In-System Programmability

The ispLSI devices are the in-system programmable version of the Lattice high density programmable Large Scale Integration (pLSI) devices. By integrating all the high voltage programming circuitry on-chip, the programming can be accomplished by simply shifting data into the device. Once the function is programmed, the non-volatile E^2CMOS cells will not lose the pattern even when the power is turned off.

All necessary programming is done via five TTL level logic interface signals. These five signals are fed into the on-chip programming circuitry where a state machine controls the programming. The simple signals for interface include isp Enable (ispEN), Serial Data In (SDI), Serial Data Out (SDO), Serial Clock (SCLK) and Mode (MODE) control. Figure 17 illustrates the block diagram of one possible scheme of the programming interface for the isp devices. For details on the operation of the internal state machine and programming of the device please refer to the In-system programming application note.

Figure 17. isp Programming Interface

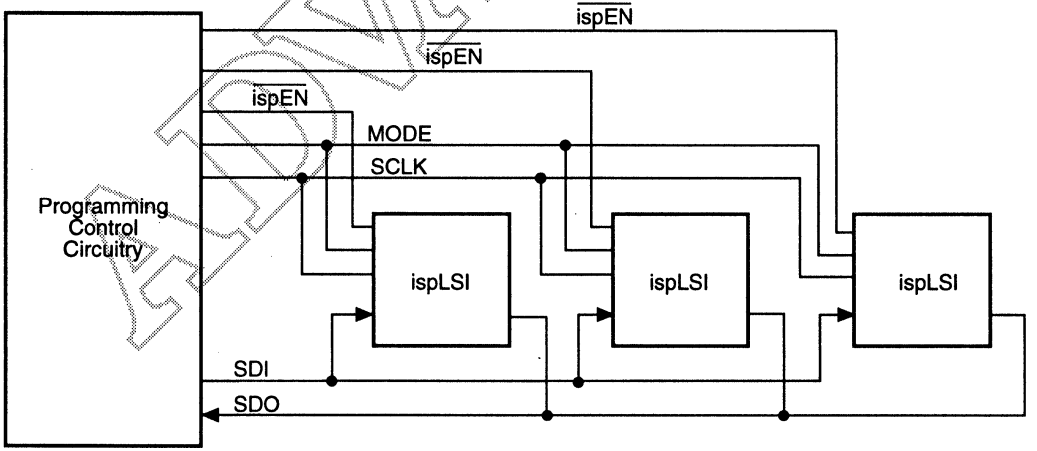
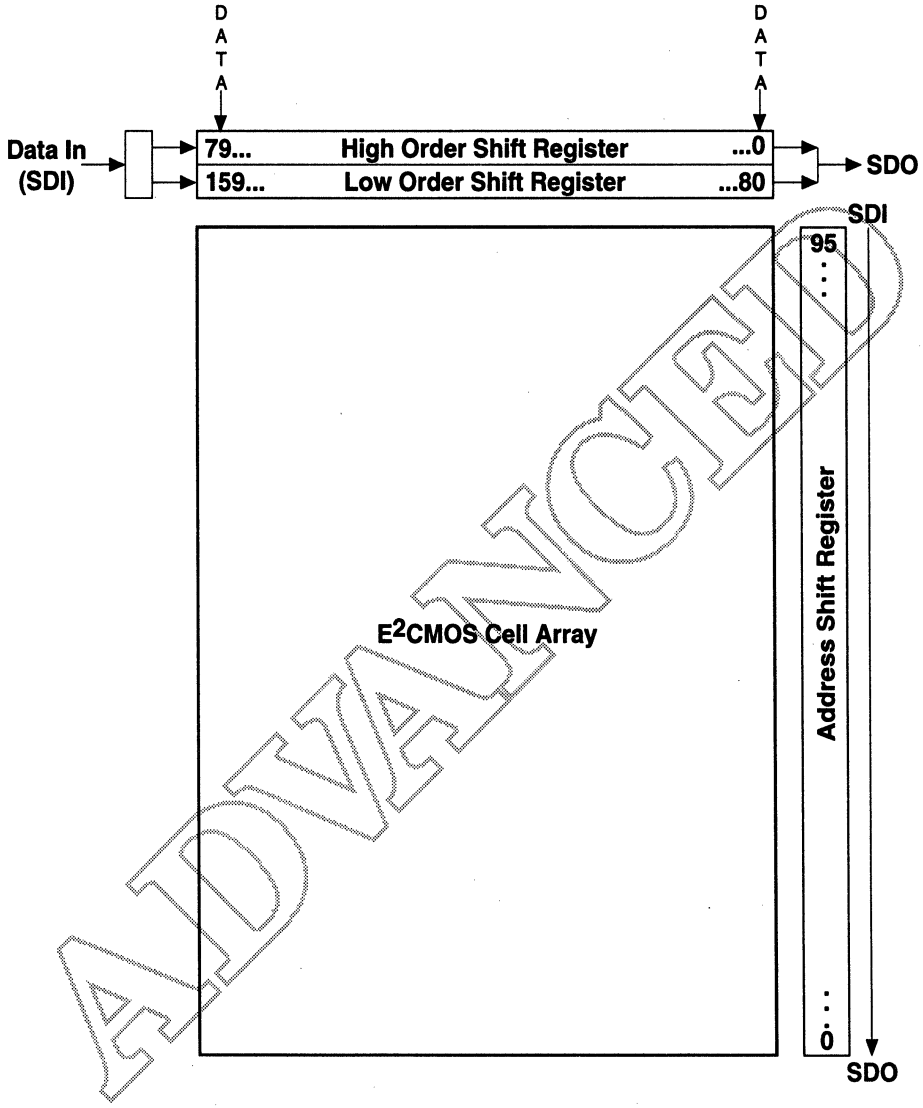


Figure 18. ispLSI Device & Shift Register Layout



Note: A logic "1" in the address shift register enables the row for programming or verification. A logic "0" disables it.

Programming Voltage/Timing Specifications

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNITS
V_{CCP}	Programming Voltage		4.75	5	5.25	V
I_{CCP}	Programming Supply Current ispEN		–	50	100	mA
V_{IHP}	Input Voltage High		2.0	–	V_{CCP}	V
V_{ILP}	Input Vopltage Low		0	–	0.8	V
I_{IP}	Input Current		–	100	200	µA
V_{OHP}	Output Volatge High	I_{OH} = -3.2 mA	2.4	–	V_{CCP}	V
V_{OLP}	Output Voltage Low	I_{OL} =5 mA	0	–	0.5	V
t_d	Pulse Sequence Delay		1	5	10	µs
t_{isp}	$\overline{ispEN}$ to Output 3-State		–	1	10	µs
t_{su}	Setup Time		.1	.5	–	µs
t_h	Hold Time		.1	.5	–	µs
t_{clk}	Clock Pulse Width		0.5	1	–	µs
t_{pwv}	Verify Pulse Width		20	30	–	µs
t_{pwp}	Programming Pulse Width		40	–	100	ms
t_{bew}	Bulk Erase Pulse Width		200	–	–	ms
t_{rst}	Reset Time From Valid V_{CCP}		45	–	–	µs

Figure 19. Timing Waveform for isp Operation

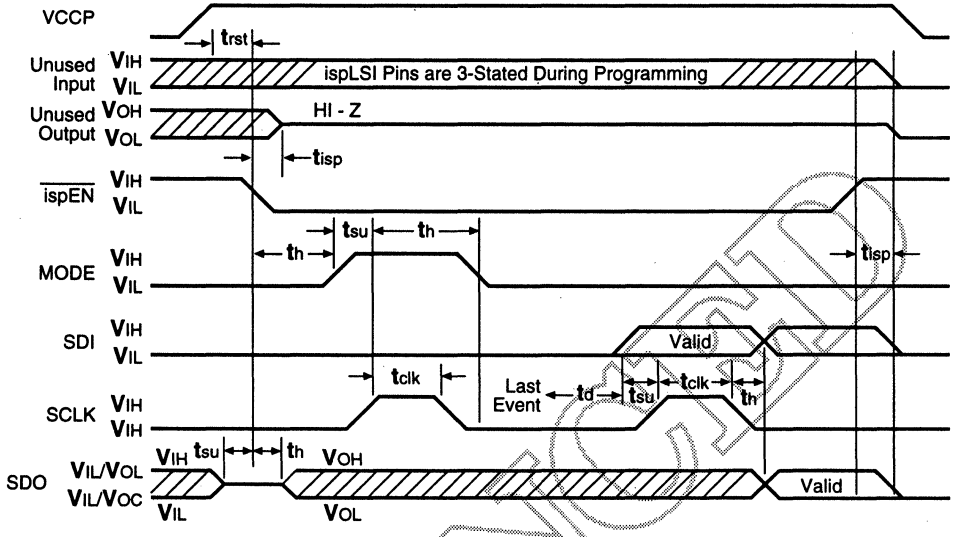
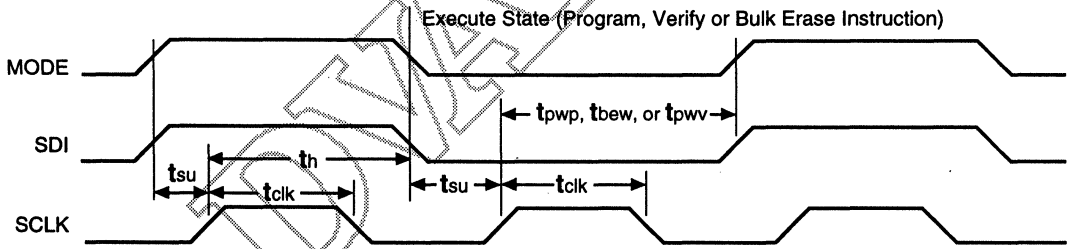
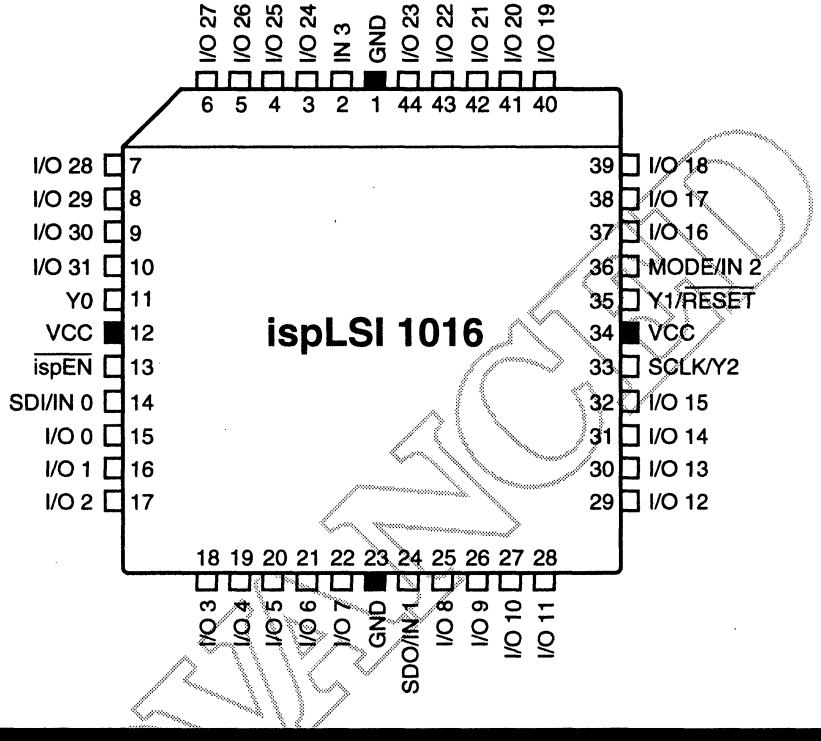


Figure 20. Program, Verify & Bulk Erase Waveform

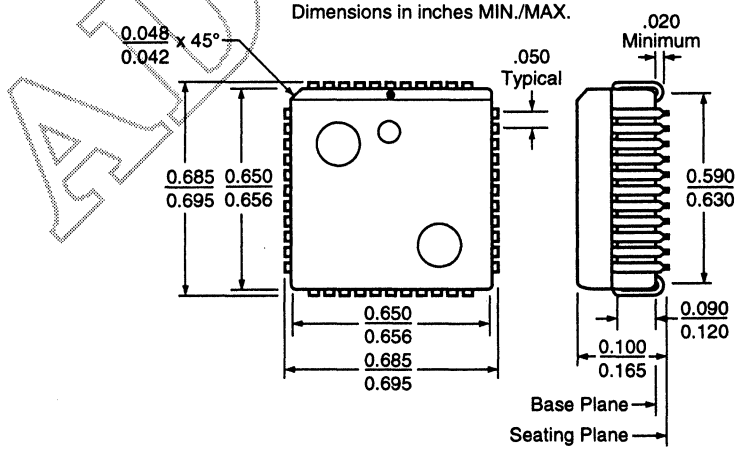


Pin Configuration

ispLSI 1016 PLCC Pinout Diagram



44-Pin PLCC



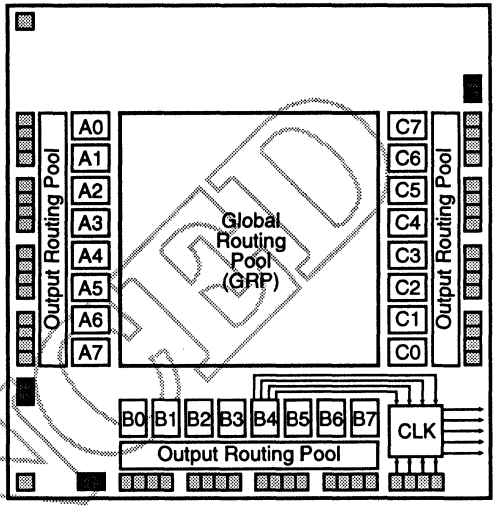
ispLSI™ 1024

in-system programmable Large Scale Integration

Features

- **In-system programmable HIGH DENSITY LOGIC**
  - **— Member of Lattice's ispLSI Family**
  - **— Fully Compatible with Lattice's pLSI™ Family**
  - **— High Speed Global Interconnects**
  - **— 48 I/O Pins, Eight Dedicated Inputs**
  - **— 144 Registers**
  - **— Wide Input Gating for Fast Counters, State Machines, Address Decoders, etc.**
  - **— Small Logic Block Size for Random Logic**
  - **— Security Cell Prevents Unauthorized Copying**
- **HIGH PERFORMANCE E²CMOS® TECHNOLOGY**
  - **— f_{max} = 80 MHz Maximum Operating Frequency**
  - **— t_{pd} = 15 ns Propagation Delay**
  - **— TTL Compatible Inputs and Outputs**
- **In-system programmable 5-VOLT ONLY**
  - **— Change Logic and Interconnects "on-the-fly" in Seconds**
  - **— Reprogram Soldered Device for Debugging**
  - **— Non-Volatile E²CMOS Technology**
  - **— 100% Tested**
- **COMBINES EASE OF USE AND THE FAST SYSTEM SPEED OF PLDs WITH THE DENSITY AND FLEXIBILITY OF FIELD PROGRAMMABLE GATE ARRAYS**
  - **— Complete Programmable Device can Combine Glue Logic and Structured Designs**
  - **— 100% Routable at 80% Utilization**
  - **— Four Dedicated Clock Input Pins**
  - **— Synchronous and Asynchronous Clocks**
  - **— Flexible Pin Placement**
  - **— Optimized Global Routing Pool Allows Global Interconnectivity**
- **pLSI/ispLSI DEVELOPMENT SYSTEM (pDS™)**
  - **— Boolean Logic Compiler**
  - **— Automatic Place and Route**
  - **— Manual Partitioning**
  - **— PC Platform**
  - **— Easy to Use Windows Interface**
- **ADVANCED pLSI/ispLSI DEVELOPMENT SYSTEM**
  - **— Industry Standard, Third Party Design Environments**
  - **— Schematic Capture**
  - **— Fully Automatic Partitioning**
  - **— Automatic Place and Route**
  - **— Comprehensive Logic and Timing Simulation**
  - **— PC and Workstation Platforms**

Functional Block Diagram



Description

The Lattice ispLSI 1024 is a High Density Programmable Logic Device featuring 5-Volt in-system programmability and in-system diagnostic capabilities. The device contains 144 Registers, 48 Universal I/O pins, 6 Dedicated Input Pins, 4 Dedicated Clock Input Pins and a Global Routing Pool (GRP). The GRP provides complete interconnectivity between all of these elements. It is the first device which offers non-volatile "on-the-fly" reprogrammability of the logic, as well as the interconnects to provide truly reconfigurable systems. It is architecturally and parametrically compatible to the pLSI 1024 device, but multiplexes four of the dedicated input pins to control in-system programming.

The basic unit of logic on the ispLSI 1024 device is the Generic Logic Block (GLB). The GLBs are labeled A0, A1 .. C7 (see figure 1). There are a total of 24 GLBs in the ispLSI 1024 device. Each GLB has 18 inputs, a programmable AND/OR/Exclusive OR array, and four outputs which can be configured to be either combinatorial or registered. Inputs to the GLB come from the GRP. All of the GLB outputs are brought back into the GRP so that they can be connected to the inputs of any other GLB on the device.

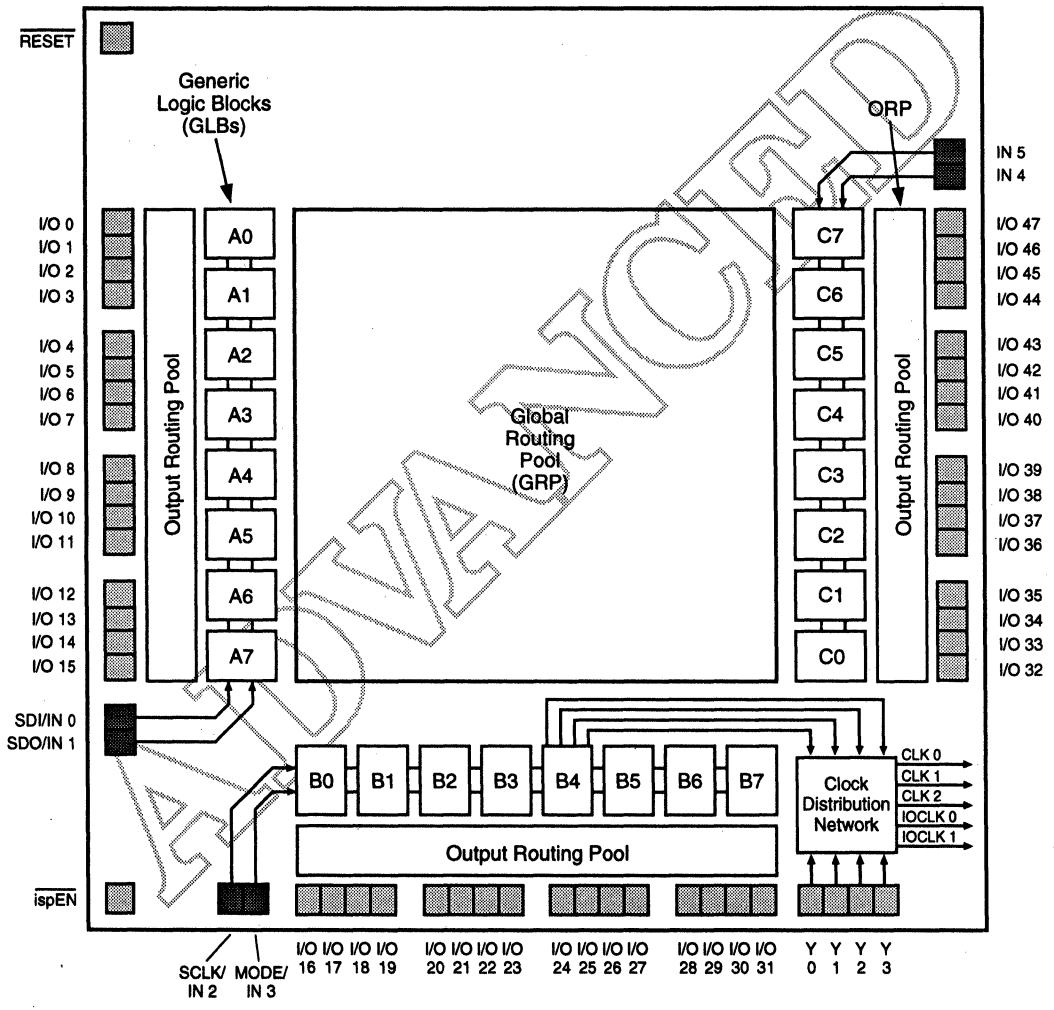
January 1992. Rev. A

LATTICE SEMICONDUCTOR CORP., 5555 Northeast Moore Ct., Hillsboro, Oregon 97124, U.S.A.
Tel. 1-800-LATTICE (528-8423); FAX (503) 681-3037

The device also has 48 I/O Cells, each of which is directly connected to an I/O pin. Each I/O cell can be individually programmed to be a combinatorial input, registered input, latched input, output or bi-directional I/O pin with 3-state. The signal levels are TTL compatible voltages and the output drivers can source 4 mA or sink 8 mA.

Functional Block Diagram

Figure 1. ispLSI 1024



Description (continued)

The 48 I/O Cells are grouped into three sets of 16 each. as shown in figure 1. Each of these I/O groups is associated with a logic Megablock through the use of the Output Routing Pool (ORP) and shares a common Output Enable (OE) signal.

Eight GLBs, 16 I/O Cells and one ORP are connected together to make a logic Megablock. The Megablock is defined by the resources that it shares. The outputs of the eight GLBs are connected to a set of 16 universal I/O cells by the ORP. The ispLSI 1024 Device contains four of these Megablocks.

The GRP has as its inputs the outputs from all of the GLBs and all of the inputs from the bi-directional I/O cells. All of these signals are made available to the inputs of the GLBs. Delays through the GRP have been equalized to minimize timing skew and logic glitching.

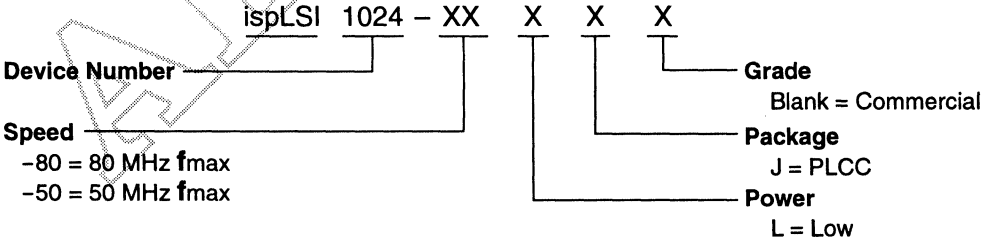
Clocks in the ispLSI 1024 device are selected using the Clock Distribution Network. Three dedicated clock pins (Y0 to Y3) are brought into the distribution network, and five outputs (CLK 0 to CLK 2 and IOCLK 0, IOCLK 1) are provided to route clocks to the GLBs and I/O cells. The Clock Distribution Network can also be driven from a special GLB (B4 on the ispLSI 1024 device). The logic of this GLB allows the user to create an internal clock from a combination of internal signals within the device.

The ispLSI 1024 device is part of Lattice's in-system programmable Large Scale Integration (ispLSI) family. This family contains a range of devices from the ispLSI 1016, with 96 registers, to the ispLSI 1048 with 288 registers. The ispLSI Family Product Selector Guide below lists key attributes of the devices along with the number of resources available.

ispLSI Family Product Selector Guide

DEVICE	ispLSI 1016	ispLSI 1024	ispLSI 1032	ispLSI 1048
GLBs	16	24	32	48
Registers	96	144	192	288
I/O Pins	32	48	64	96
Dedicated Inputs	4	6	8	10
Pin Count	44	68	84	120

Ordering Information



Pin Description

Name	PLCC Pin Numbers	Description
I/O 0 - I/O 3 I/O 4 - I/O 7 I/O 8 - I/O 11 I/O 12 - I/O 15 I/O 16 - I/O 19 I/O 20 - I/O 23 I/O 24 - I/O 27 I/O 28 - I/O 31 I/O 32 - I/O 35 I/O 36 - I/O 39 I/O 40 - I/O 43 I/O 44 - I/O 47	22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14	Input/Output Pins - These are the general purpose I/O pins used by the logic array.
IN 4 - IN 5	2, 15	Dedicated input pins to the device.
$\overline{\text{ispEN}}$	19	Input – Dedicated in-system programming enable input pin. This pin is brought low to enable the programming mode. The MODE, SDI, SDO and SCLK options become active.
SDI/IN 0	21	Input – This pin performs two functions. It is a dedicated input pin when $\overline{\text{ispEN}}$ is logic high. When $\overline{\text{ispEN}}$ is logic low, it functions as an input pin to load programming data into the device. SDI/IN 0 also is used as one of the two control pins for the isp state machine.
MODE/IN 3	55	Input – This pin performs two functions. It is a dedicated input pin when $\overline{\text{ispEN}}$ is logic high. When $\overline{\text{ispEN}}$ is logic low, it functions as a pin to control the operation of the isp state machine.
SDO/IN 1	34	Input/Output – This pin performs two functions. It is a dedicated clock input pin when $\overline{\text{ispEN}}$ is logic high. When $\overline{\text{ispEN}}$ is logic low, it functions as an output pin to read serial shift register data.
SCLK/IN 2	49	Input – This pin performs two functions. It is a dedicated input when $\overline{\text{ispEN}}$ is logic high. When $\overline{\text{ispEN}}$ is logic low, it functions as a clock pin for the Serial Shift Register.
$\overline{\text{RESET}}$	20	Active Low (0) Reset pin which resets all of the GLB and I/O registers in the device.
Y0	16	Dedicated Clock input. This clock input is connected to one of the clock inputs of all of the GLBs on the device.
Y1	54	Dedicated clock input. This clock input is brought into the clock distribution network, and can optionally be routed to any GLB on the device.
Y2	51	Dedicated clock input. This clock input is brought into the clock distribution network, and can optionally be routed to any GLB and/or any I/O Cell on the device.
Y3	50	Dedicated clock input. This clock input is brought into the clock distribution network, and can optionally be routed to any I/O Cell on the device.
GND	1, 18, 35, 52	Ground (GND)
VCC	17, 36, 53, 68	V_{CC}

Absolute Maximum Ratings [1]

Supply Voltage V_{CC} . . . . . -0.5 to +7.0V

Input Voltage Applied. . . . . .-2.5 to V_{CC} +1.0V

Off-State Output Voltage Applied . . . . . .-2.5 to V_{CC} +1.0V

Storage Temperature . . . . . .-65 to 125°C

Ambient Temp. with Power Applied . . . . . .-55 to 125°C

1. Stresses above those listed under the "Absolute Maximum Ratings" may cause permanent damage to the device. Functional operation of the device at these or at any other conditions above those indicated in the operational sections of this specification is not implied (while programming, follow the programming specifications).

DC Recommended Operating Condition

SYMBOL	PARAMETER	MIN.	MAX.	UNITS
T_A	Ambient Temperature	0	70	°C
V_{CC}	Supply Voltage	4.75	5.25	V
V_{IL}	Input Low Voltage	0	0.8	V
V_{IH}	Input High Voltage	2.0	V_{CC}	V

Capacitance (T_A=25°C,f=1.0 MHz)

SYMBOL	PARAMETER	MAXIMUM[1]	UNITS	TEST CONDITIONS
C_1	Input Capacitance	8	pf	V_{CC}=5.0V, V_{IN}=2.0V
C_2	I/O, Y Capacitance	10	pf	V_{CC}=5.0V, VI/O, Y=2.0V

1. Guaranteed but not 100% tested.

Data Retention Specifications

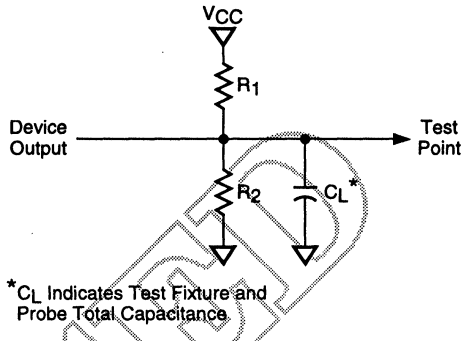
PARAMETER	MINIMUM	MAXIMUM	UNITS
Data Retention	20	–	YEARS
Erase/Reprogram Cycles	–	1000	CYCLES

Switching Test Conditions

Input Pulse Levels	GND to 3.0V
Input Rise and Fall Times	3ns 10% to 90%
Input Timing Reference Levels	1.5V
Output Timing Reference Levels	1.5V
Output Load	See Figure 2

3-state levels are measured 0.5V from steady-state active level.

Figure 2. Test Load



Output Load Conditions (see figure 2)

Test Condition		R1	R2	CL
1		470Ω	390Ω	35pF
2	Active High	∞	390Ω	35pF
	Active Low	470Ω	390Ω	35pF
3	Active High to Z at V_{OH} - 0.5V	∞	390Ω	5pF
	Active Low to Z at V_{OL} + 0.5V	470Ω	390Ω	5pF

DC Electrical Characteristics

Over Recommended Operating Conditions

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNITS
V_{OL}	Output Low Voltage	I_{OL} =8 mA.	–	–	0.4	V
V_{OH}	Output High Voltage	I_{OH} =-4 mA.	2.4	–	–	V
I_{IL}	Input or I/O Low Leakage Current	$0V \le V_{IN} \le V_{IL}$ (MAX.)	–	–	-10	µA
I_{IH}	Input or I/O High Leakage Current	$V_{IH} \le V_{IN} \le V_{CC}$	–	–	10	µA
I_{OS}[1]	Output Short Circuit Current	V_{CC} = 5V, V_{OUT} = 0.5V	-60	–	-200	mA
I_{CC}[2]	Operating Power Supply Current	V_{IL} = 0.5V, V_{IH} = 3.0V F_{TOGGLE} = 20 MHz	–	–	–	mA

1. One output at a time for a maximum duration of one second. (Vout = 0.5V)
2. Measured at a frequency of 20 MHz using six 16-bit counters.

External Switching Characteristics[1, 2, 3] — ispLSI 1024-80

Over Recommended Operating Conditions

PARAMETER	TEST[6] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
t_{pd1}	1	1	Data Propagation Delay, 4PT bypass, ORP bypass	–	12	15	ns
t_{pd2}	1	2	Data Propagation Delay	–	15	20	ns
t_{co1}[5]	1	3	External Clock to Output Delay, ORP bypass	–	8	11	ns
t_{co2}[5]	1	4	External Clock to Output Delay	–	9	14	ns
t_{co3}	1	5	Internal Synch. Clock to Output Delay	–	15	20	ns
t_{co4}	1	6	Asynchronous Clock to Output Delay	–	13	20	ns
t_{r1}	1	7	External Pin Reset to Output Delay	–	13	20	ns
t_{r2}	1	8	Asynchronous PT Reset to Output Delay	–	15	22	ns
t_{en}	2	9	Input to Output Enable	–	13	20	ns
t_{dis}	3	10	Input to Output Disable	–	13	20	ns

External AC Recommended Operating Conditions[1, 2, 3] — ispLSI 1024-80

Over Recommended Operating Conditions

PARAMETER	TEST[6] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
f_{max}[4]	1	11	Clock Frequency with Internal Feedback	–	100	80	MHz
f_{max} (External)	1	12	Clock Frequency with External Feedback	–	70	50	MHz
t_{su1}	–	13	Setup Time before External Synch. Clock, 4PT bypass	9	6	–	ns
t_{su2}	–	14	Setup Time before External Synch Clock	12	8	–	ns
t_{su3}	–	15	Setup Time before Internal Synch. Clock	9	3	–	ns
t_{su4}	–	16	Setup Time before Asynchronous Clock	9	4	–	ns
t_{h1}	–	17	Hold time after External Synchronous Clock, 4PT bypass	2	-1	–	ns
t_{h2}	–	18	Hold time after External Synchronous Clock	2	-1	–	ns
t_{h3}	–	19	Hold time after Internal Synchronous Clock	8	2	–	ns
t_{h4}	–	20	Hold time after Asynchronous Clock	8	1	–	ns
t_{rw1}	–	21	External Reset Pulse Duration	10	8	–	ns
t_{rw2}	–	22	Asynchronous Reset Pulse Duration	10	8	–	ns
t_{wh1}, t_{wl1}	–	23,24	External Synchronous Clock Pulse Duration, High, Low	6	5	–	ns
t_{wh2}, t_{wl2}	–	25,26	Asynchronous Clock Pulse Duration, High, Low	6	5	–	ns

1. External Parameters are tested and guaranteed.
2. See Timing Technical Note for further details.
3. Unless noted, all parameters use ORP, GRP fanout of four, PTSA, and are measured with 16 outputs switching.
4. Standard 16-bit counter implementation using GRP feedback.
5. Clock to output specifications include a maximum skew of 2 ns.
6. Refer to Switching Test Conditions section.

Switching Characteristics[1, 2, 3] — ispLSI 1024-80

Using I/O Cell

PARAMETER	TEST[4] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
t_{su5}	–	27	Setup Time before External Synchronous Clock	5	0	–	ns
t_{su6}	–	28	Setup Time before Internal Synchronous Clock	0	-3	–	ns
t_{h5}	–	29	Hold Time after External Synchronous Clock	8	4	–	ns
t_{h6}	–	30	Hold Time after Internal Synchronous Clock	15	11	–	ns
t_{wh3}, t_{wl3}	–	31,32	Clock Pulse Duration, High, Low	6	5	–	ns

1. External Parameters are tested and guaranteed.
2. See Timing Technical Note for further details.
3. Unless noted, all parameters use ORP, GRP fanout of four, PTSA, and are measured with 16 outputs switching.
4. Refer to Switching Test Conditions section.

External Switching Characteristics[1, 2, 3] — ispLSI 1024-50

Over Recommended Operating Conditions

PARAMETER	TEST[6] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
t_{pd1}	1	1	Data Propagation Delay, 4PT bypass, ORP bypass	–	16	20	ns
t_{pd2}	1	2	Data Propagation Delay, ORP	–	19	25	ns
t_{co1}[5]	1	3	External Clock to Output Delay, ORP bypass	–	12	16	ns
t_{co2}[5]	1	4	External Clock to Output Delay	–	15	20	ns
t_{co3}	1	5	Internal Synch. Clock to Output Delay	–	21	28	ns
t_{co4}	1	6	Asynchronous Clock to Output Delay	–	21	28	ns
t_{r1}	–	7	External Pin Reset to Output Delay	–	21	28	ns
t_{r2}	–	8	Asynchronous PT Reset to Output Delay	–	24	30	ns
t_{en}	2	9	Input to Output Enable	–	21	28	ns
t_{dis}	3	10	Input to Output Disable	–	21	28	ns

External AC Recommended Operating Conditions[1, 2, 3] — ispLSI 1024-50

Over Recommended Operating Conditions

PARAMETER	TEST[6] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
f_{max}[4]	1	11	Clock Frequency with Internal Feedback	–	70	50	MHz
f_{max} (External)	1	12	Clock Frequency with External Feedback	–	45	33	MHz
t_{su1}	–	13	Setup Time before External Synch. Clock, 4PT bypass	14	10	–	ns
t_{su2}	–	14	Setup Time before External Synch Clock	17	13	–	ns
t_{su3}	–	15	Setup Time before Internal Synch. Clock	13	9	–	ns
t_{su4}	–	16	Setup Time before Asynchronous Clock	13	9	–	ns
t_{h1}	–	17	Hold time after External Synchronous Clock, 4PT bypass	7	3	–	ns
t_{h2}	–	18	Hold time after External Synchronous Clock	7	3	–	ns
t_{h3}	–	19	Hold time after Internal Synchronous Clock	11	5	–	ns
t_{h4}	–	20	Hold time after Asynchronous Clock	11	5	–	ns
t_{rw1}	–	21	External Reset Pulse Duration	15	13	–	ns
t_{rw2}	–	22	Asynchronous Reset Pulse Duration	15	13	–	ns
t_{wh1}, t_{wl1}	–	23,24	External Synchronous Clock Pulse Duration, High, Low	10	8	–	ns
t_{wh2}, t_{wl2}	–	25,26	Asynchronous Clock Pulse Duration, High, Low	10	8	–	ns

1. External Parameters are tested and guaranteed.
2. See Timing Technical Note for further details.
3. Unless noted, all parameters use ORP, GRP fanout of four, PTSA, and are measured with 16 outputs switching.
4. Standard 16-bit counter implementation using GRP feedback.
5. Clock to output specifications include a maximum skew of 2 ns.
6. Refer to Switching Test Conditions section.

Switching Characteristics[1, 2, 3] — ispLSI 1024-50

Using I/O Cell

PARAMETER	TEST[4] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
t_{su5}	–	27	Setup Time before External Synchronous Clock	10	5	–	ns
t_{su6}	–	28	Setup Time before Internal Synchronous Clock	0	-5	–	ns
t_{h5}	–	29	Hold Time after External Synchronous Clock	12	6	–	ns
t_{h6}	–	30	Hold Time after Internal Synchronous Clock	20	15	–	ns
t_{wh3}, t_{wl3}	–	31,32	Clock Pulse Duration, High, Low	10	8	–	ns

1. External Parameters are tested and guaranteed.
2. See Timing Technical Note for further details.
3. Unless noted, all parameters use ORP, GRP fanout of four, PTSA, and are measured with 16 outputs switching.
4. Refer to Switching Test Conditions section.

Architectural Description

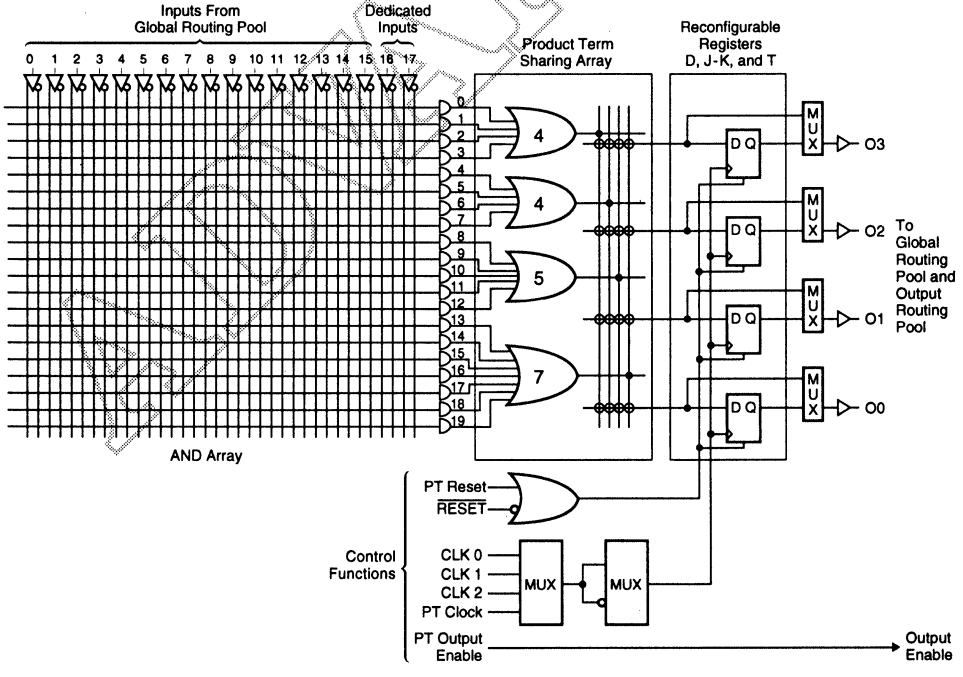
The Generic Logic Block

The Generic Logic Block (GLB) is the standard logic block of the Lattice High Density ispLSI Device. This GLB has 18 inputs, four outputs and the logic necessary to implement most standard logic functions. The internal logic of the GLB is divided into four separate sections (see figure 3). The AND Array, the Product Term Sharing Array (PTSA), the Reconfigurable Registers, and the Control Functions. The AND array consists of 20 product terms which can produce the logical sum of any of the 18 GLB inputs. Sixteen of the inputs come from the Global Routing Pool, and are either feedback signals from any of the 24 GLBs or inputs from the external I/O Cells. The two remaining inputs come directly from two dedicated input pins. These signals are available to the product terms in both the logical true and the complemented forms which makes boolean logic reduction easier.

The PTSA takes the 20 product terms and allocates them to the four GLB outputs. There are four OR gates, with four, four, five and seven product terms (see figure 3). The output of any of these gates can be routed to any of the four GLB outputs, and if more product terms are needed, the PTSA can combine them as necessary. If the users' main concern is speed, the PTSA can use a bypass circuit with four product terms to increase the performance of the cell (see figure 4). This can be done to any or all of the four outputs from the GLB.

The Reconfigurable Registers consist of four D-type flip-flops with an Exclusive OR Gate on the input. The Exclusive OR gate in the GLB can be used either as a logic element or to reconfigure the D-type flip-flop to emulate a J-K, or T-type flip-flop (see figure 5). This greatly simplifies the design of counters, comparators and ALU type functions. The registers can be bypassed if the user needs a combinatorial output. Each register output is brought back into the Global Routing Pool and is also brought to the I/O cells via the Output Routing Pool. Reconfigurable registers are not available when the four product term bypass is used.

Figure 3. GLB: Product Term Sharing Array



Architectural Description

Figure 4. GLB: Four Product Term Bypass

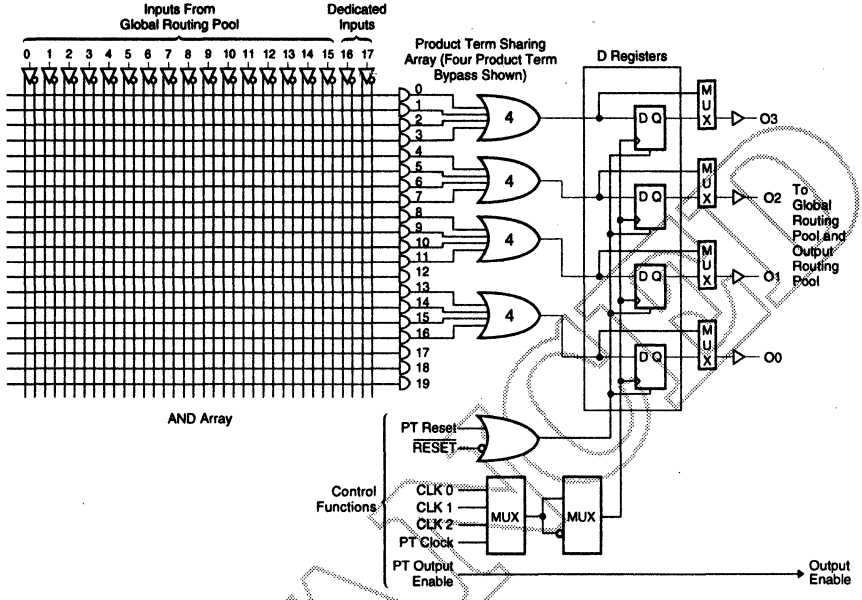
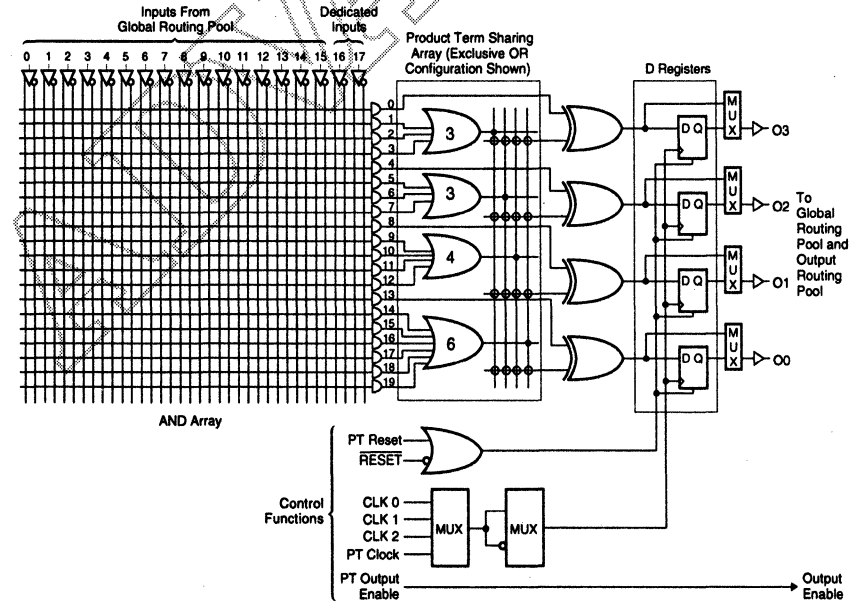


Figure 5. GLB: Exclusive OR Gate



Architectural Description

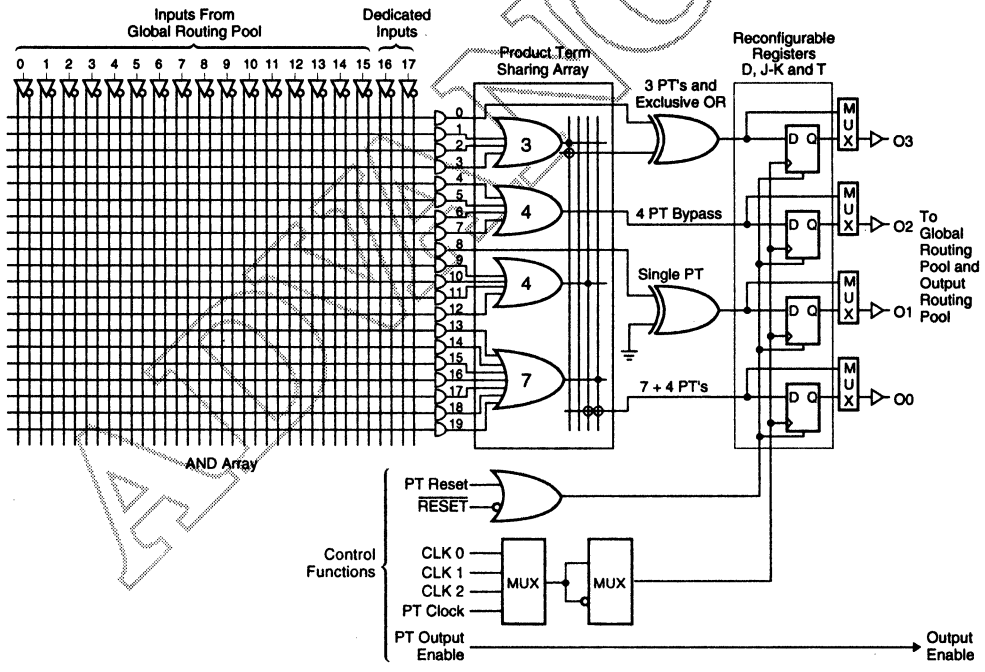
The Generic Logic Block (continued)

The PTSA is flexible enough to allow these features to be used in virtually any combination that the user desires. In the GLB shown in figure 6, Output Three (O3) is configured using the XOR Gate and Output Two (O2) is configured using the four Product Term Bypass. Output One (O1) uses one of the inputs from the five Product Term OR gate while Output Zero (O0) combines the remaining four product terms with all of the product terms from the seven Product Term OR gate for a total of eleven (7+4).

Various signals which control the operation of the GLB are developed in the Control Function section. The clock for the registers can come from any of three sources developed in the Clock Distribution Network (See Clock Distribution Network Section) or from a product term within the GLB. The Reset Signal for the GLB can come from the Global Reset pin or from a product term within the block. The Output Enable for the I/O cells associated with the GLB comes from a product term within the block. Use of a product term for a control function makes that product term unavailable for use as a logic term. Refer to the following table to determine which logic functions are affected.

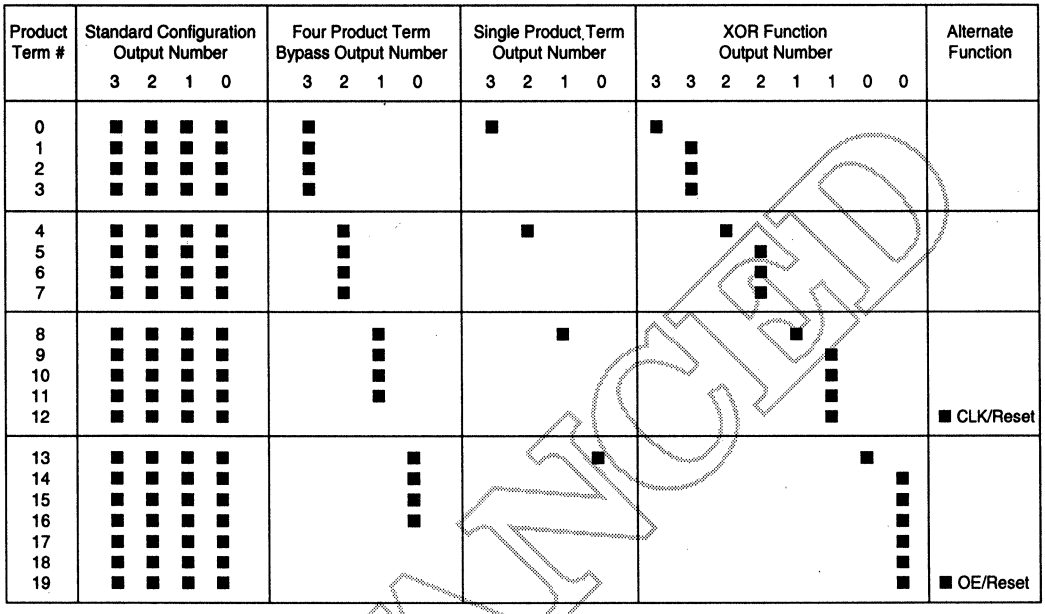
There are additional features in a GLB which allow implementation of logic intensive functions. These features are accessible using the hard Macros from the software and require no intervention on the part of the user.

Figure 6. GLB: Various Logical Combinations



Architectural Description

Product Term Sharing Matrix



This matrix shows how each of the product terms are used in the various modes. As an example, Product Term 12 can be used as an input to the five input OR gate in the standard configuration. This OR gate under standard configuration can be routed to any of the four GLB outputs. Product Term 12 is not used in the four product term bypass mode. When GLB output one is used in the Exclusive OR (XOR) mode Product Term 12 becomes one of the inputs to the four input OR Gate. If Product Term 12 is not used in the logic, then it is available for use as either the Asynchronous Clock signal or the GLB Reset signal.

Architectural Description

The Megablock

The ispLSI family is structured into multiple "Megablocks". A Megablock consists of 8 GLBs, an Output Routing Pool (ORP) and 16 I/O Cells. Each of these will be explained in detail in the following sections. These elements are coupled together as shown in figure 7. This logic structure is referred to as the Megablock. The various members of the ispLSI family are created by combining from two to six Megablocks on a single device.

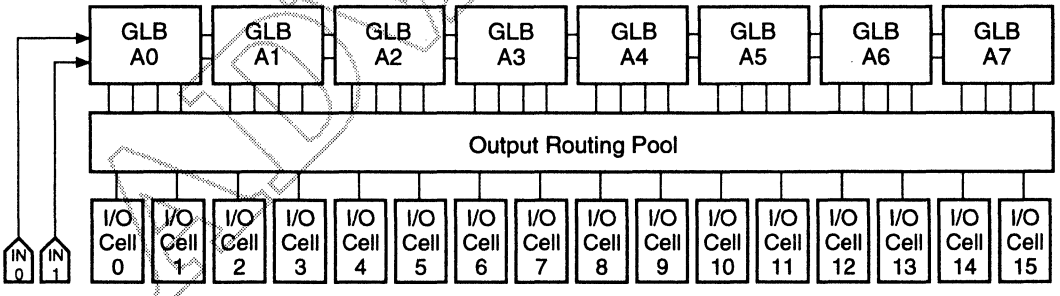
The Megablock shares two sets of resources. The eight GLBs within the Megablock share two dedicated input pins. These dedicated input pins are not available to GLBs in any other Megablock. These pins are dedicated (non-registered) inputs only and are automatically assigned by software. One Output Enable signal is generated within the Megablock and is common to all sixteen of the I/O Cells in the Megablock. The Output Enable signal can be generated using a product term in any of the eight GLBs within the Megablock (See the following section on the Output Enable Multiplexers).

Because of the shared logic within the Megablock, signals which share a common function (counters, busses, etc.) should be grouped within a Megablock. This will allow the user to obtain the best utilization of the logic within the device and eliminate routing bottlenecks.

Megablocks in Each Device

DEVICE	MEGABLOCKS	GLBs	I/O CELLS
ispLSI 1016	2	16	32
ispLSI 1024	3	24	48
ispLSI 1032	4	32	64
ispLSI 1048	6	48	96

Figure 7. The Megablock



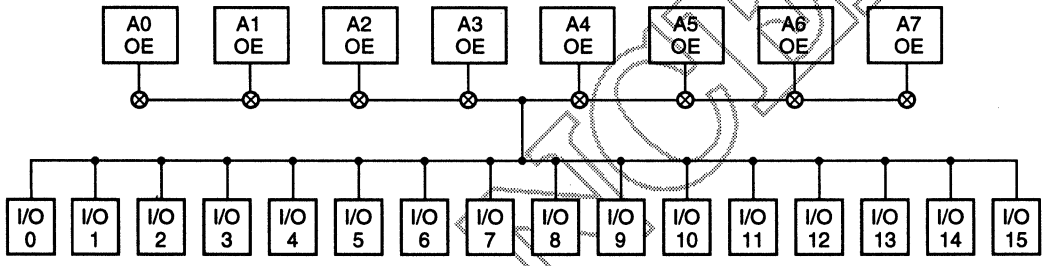
Note: The inputs from the I/O Cell are not shown in this figure, they go directly to the GRP.

Architectural Description

The Output Enable Control

One Output Enable (OE) signal can be generated within each GLB using the OE Product Term 19. One of the eight OE signals from each GLB within a Megablock, is then routed to all of the I/O Cells within the Megablock (see figure 8). This OE signal can simultaneously control all of the 16 I/O cells which are used in 3-state mode. Individual I/O cells also have independent control for permanently enabling or disabling the output buffer (Refer to the I/O Cell Section). Only one OE signal is allowed per Megablock for 3-state operation. The advantage to this approach is that the OE signal can be generated in any GLB within the Megablock which happens to have an unused OE product term. This frees up the other OE product terms for use as logic.

Figure 8. Output Enable Controls



Architectural Description

The Output Routing Pool

The Output Routing Pool (ORP) routes signals from the Generic Logic Block outputs to I/O Cells configured as outputs or bi-directional pins (see figure 9). The purpose of the ORP is to allow greater flexibility when assigning I/O pins. It also simplifies the job for the routing software which results in a higher degree of utilization.

By examining the programmable switch matrix in figure 9, it can be seen that a GLB output can be connected to one of four I/O Cells. Further flexibility is provided by using the PTSA, (Figures 3 through 7) which makes the GLB outputs completely interchangeable. This allows the routing program to freely interchange the outputs to achieve the best routability. This is an automatic process and requires no intervention on the part of the user.

The ORP bypass connections (see figure 10) further increase the flexibility of the device. The ORP bypass connect specific GLB outputs to specific I/O Cells at a faster speed. The bypass path tends to restrict the routability of the device and should only be used for critical signals.

Figure 9. Output Routing Pool

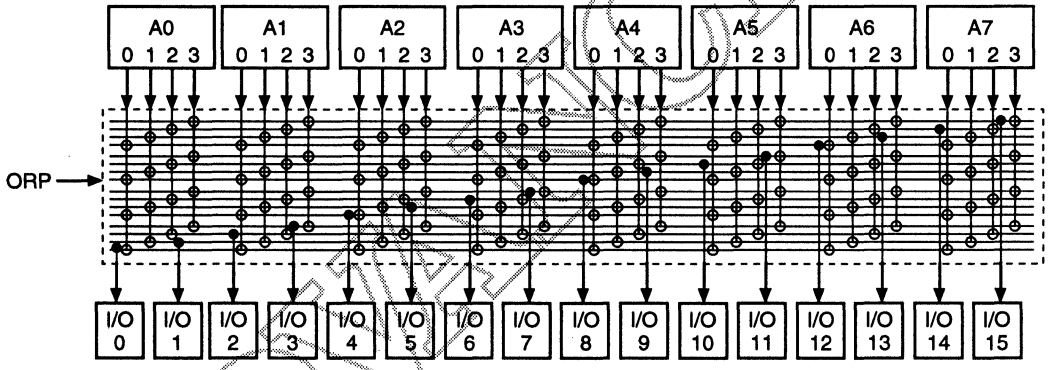
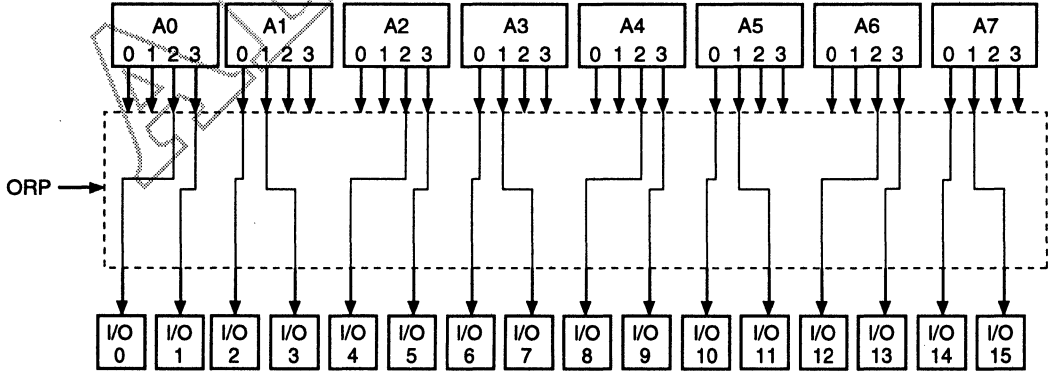


Figure 10. Output Routing Pool showing Bypass



Architectural Description

I/O Cell

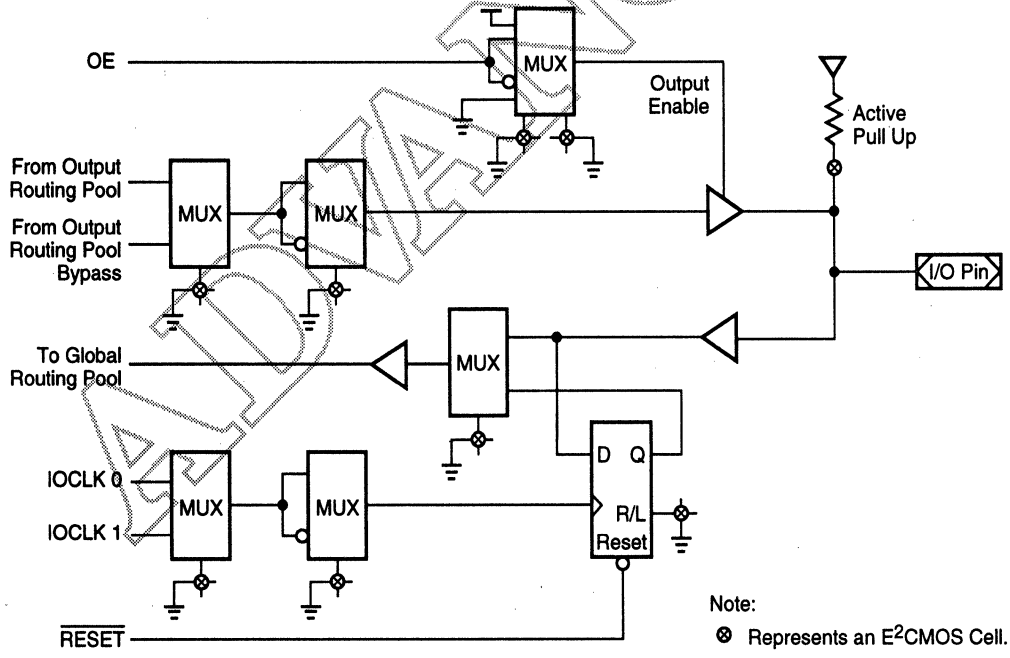
The I/O Cell (see figure 11) is used to route input, output or bidirectional signals connected to the I/O pin. The two logic inputs come from the ORP (see figure 7). One comes from the ORP, and the other comes from the faster ORP bypass. A pair of multiplexers select which signal will be used, and its polarity.

The OE signal comes from the Output Enable. As with the data path, a multiplexer selects the signal polarity. The Output Enable can be set to a logic high (Enabled) when a straight output pin is desired, or logic low (Disabled) when a straight input pin is needed. The Global Reset ($\overline{\text{RESET}}$) signal is driven by the active low chip reset pin. Each I/O Cell can individually select one of the two clock signals (IOCLK 0 or IOCLK 1). These clock signals are generated by the Clock Distribution Network.

Using the multiplexers, the I/O Cell can be configured as an input, an output, a 3-stated output or a bidirectional I/O. The D-type register can be configured as a level sensitive transparent latch or an edge triggered flip-flop to store the in-coming data. Figure 12 illustrates some of the various I/O cell configurations possible.

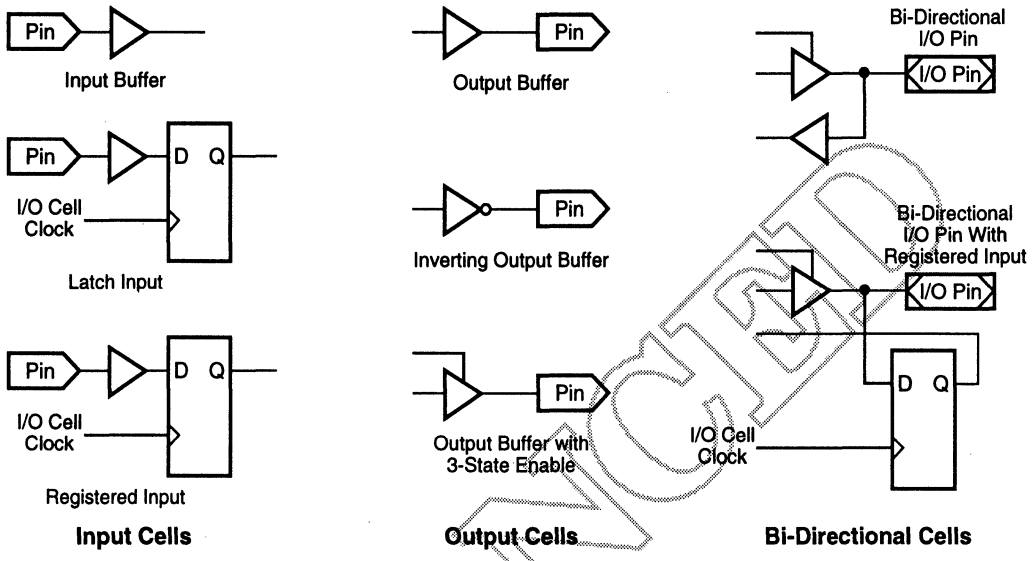
There is an Active Pullup resistor on the I/O pins which is automatically used when the pin is not connected. This prevents the pin from floating and inducing noise into the device or consuming additional power.

Figure 11. I/O Cell Architecture



Architectural Description

Figure 12. Example I/O Cell Configurations



Architectural Description

Clock Distribution Network

The Clock Distribution Network is shown in figure 13. It generates five global clock signals CLK 0, CLK 1, CLK 2 and IOCLK 0, IOCLK 1. The first three, CLK 0, CLK 1 and CLK 2 are used for clocking all the GLBs in the device. Similarly, IOCLK 0 and IOCLK 1 signals are used for clocking all of the I/O Cells in the device. There are four dedicated system clock pins (Y0, Y1, Y2, Y3) which can be directed to any GLB or any I/O Cell using the Clock Distribution Network. The other inputs to the Clock Distribution Network are the 4 outputs of a dedicated clock GLB ("B4" for ispLSI 1024). These Clock GLB outputs can be used to create a user-defined internal clocking scheme.

Typically the clock GLB will be clocked using an external main clock pin Y0 connected to global clock signal CLK 0. The outputs of the clock GLB in turn will generate "divide by two" or "divide by four" phases of the CLK 0 which can be connected to CLK 1, CLK 2 or IOCLK 0, IOCLK 1 global clocks.

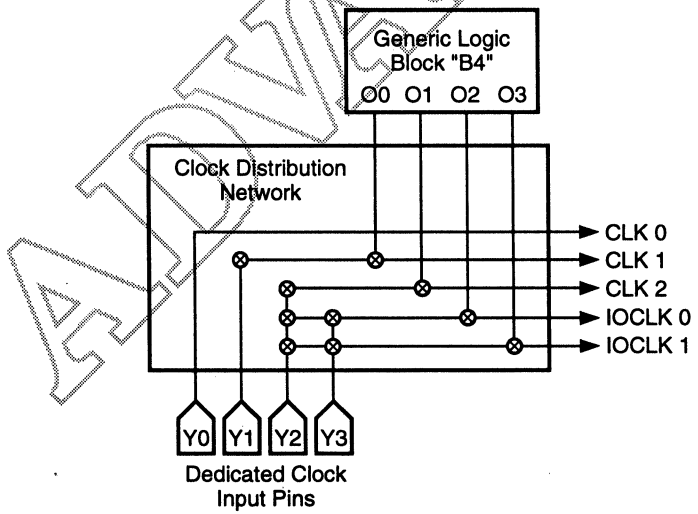
All GLBs also have the capability of generating their own asynchronous clocks using Product Term 12. CLK 0, CLK 1 and CLK 2 feed to their corresponding inputs on all the GLBs (see figure 3).

The two I/O clocks generated in the Clock Distribution Network IOCLK 0 and IOCLK 1, are brought to all 64 of the I/O cells and the user programs the I/O cell to use one of the two.

Global Routing Pool

The Global Routing Pool (GRP) is a Lattice proprietary interconnect structure which offers fast predictable speeds with complete connectivity. The GRP allows the outputs from the GLBs or the I/O cell inputs to be connected to the inputs of the GLBs. Any GLB output is available to the input of all other GLBs, and similarly an input from an I/O pin is available as an input to all of the GLBs. Because of the uniform architecture of the ispLSI device, the delays through the Global Routing Pool are both consistent and predictable. However, they are slightly affected by fanout.

Figure 13. Clock Distribution Network



Security Cell

A security cell is provided in the ispLSI devices to prevent unauthorized copying of the array patterns. Once programmed, this cell prevents further read access to the functional bits in the device. This cell can only be erased by reprogramming the device, so the original configuration can never be examined once this cell is programmed.

Device Programming

ispLSI devices are programmed using a Lattice-approved Device Programmer, available from a number of third party manufacturers or in-system using Lattice programming algorithms. Complete programming of the device takes only a few seconds. Erasing of the device is automatic and is completely transparent to the user.

Latch-up Protection

ispLSI devices are designed with an on-board charge pump to negatively bias the substrate. The negative bias is of sufficient magnitude to prevent input undershoots from causing the internal circuitry to latchup. Additionally, outputs are designed with n-channel pull-ups instead of the traditional p-channel pull-ups to eliminate any possibility of SCR induced latching.

In-System Programmability

The ispLSI devices are the in-system programmable version of the Lattice high density programmable Large Scale Integration (pLSI) devices. By integrating all the high voltage programming circuitry on-chip, the programming can be accomplished by simply shifting data into the device. Once the function is programmed, the non-volatile E^2CMOS cells will not lose the pattern even when the power is turned off.

All necessary programming is done via five TTL level logic interface signals. These five signals are fed into the on-chip programming circuitry where a state machine controls the programming. The simple signals for interface include isp Enable (ispEN), Serial Data In (SDI), Serial Data Out (SDO), Serial Clock (SCLK) and Mode (MODE) control. Figure 17 illustrates the block diagram of one possible scheme of the programming interface for the isp devices. For details on the operation of the internal state machine and programming of the device please refer to the In-system programming application note.

Figure 17. isp Programming Interface

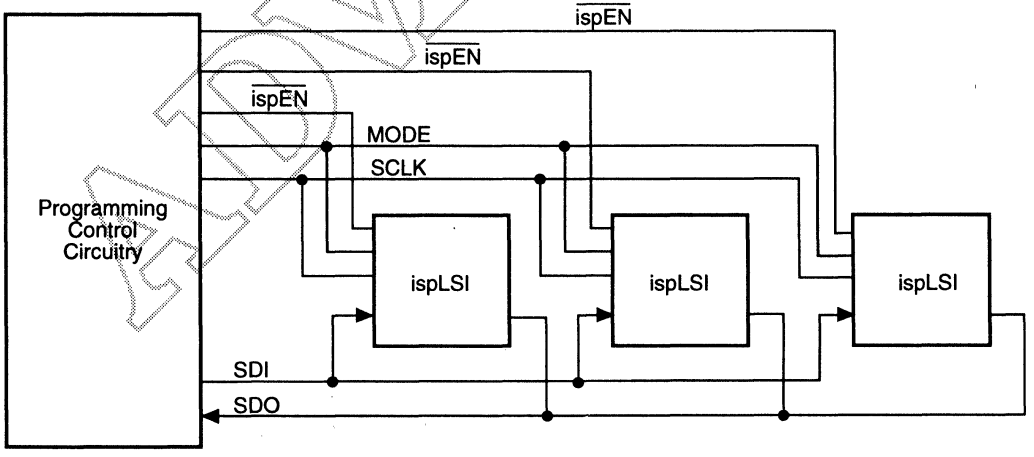
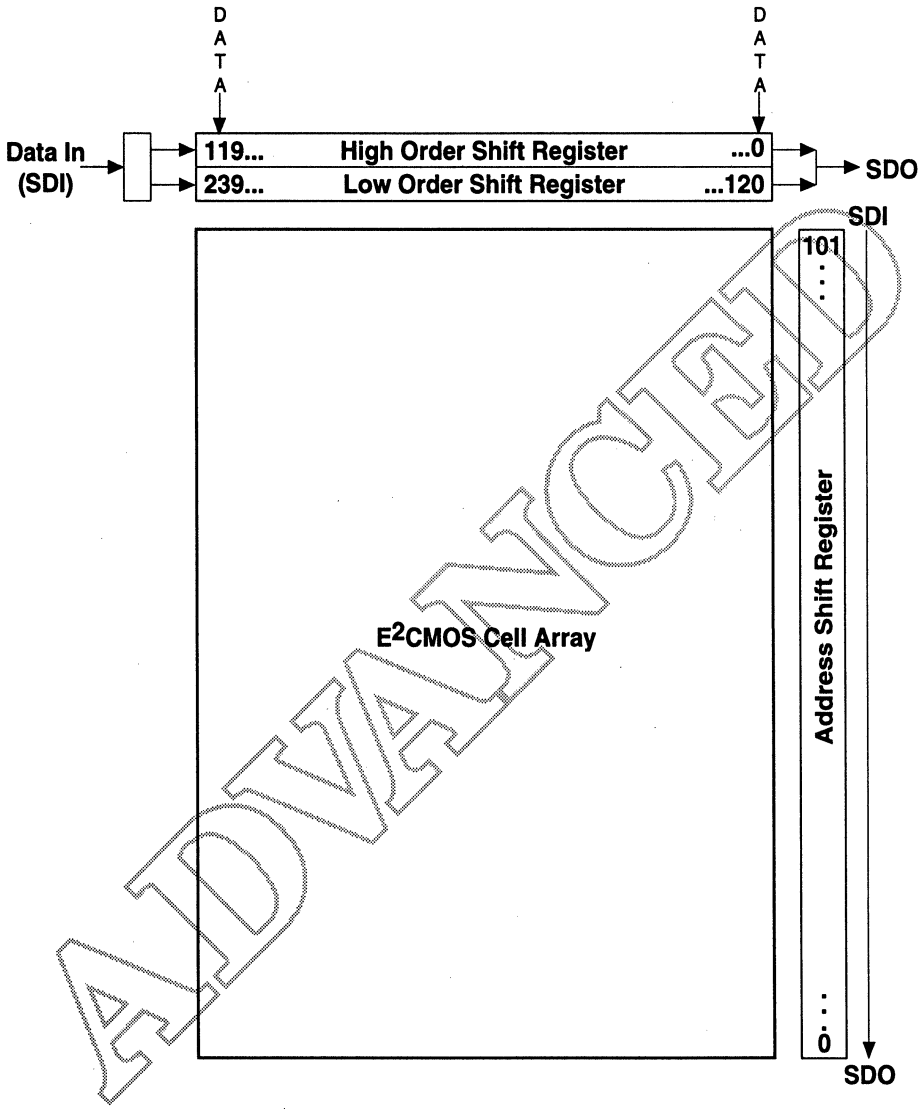


Figure 18. ispLSI Device & Shift Register Layout



Note: A logic "1" in the address shift register enables the row for programming or verification. A logic "0" disables it.

Programming Voltage/Timing Specifications

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNITS
V_{CCP}	Programming Voltage		4.75	5	5.25	V
I_{CCP}	Programming Supply Current ispEN		–	50	100	mA
V_{IHP}	Input Voltage High		2.0	–	V_{CCP}	V
V_{ILP}	Input Vopltage Low		0	–	0.8	V
I_{IP}	Input Current		–	100	200	µA
V_{OHP}	Output Volatge High	I_{OH} = -3.2 mA	2.4	–	V_{CCP}	V
V_{OLP}	Output Voltage Low	I_{OL} =5 mA	0	–	0.5	V
t_d	Pulse Sequence Delay		1	5	10	µs
t_{isp}	$\overline{ispEN}$ to Output 3-State		–	1	10	µs
t_{su}	Setup Time		1	.5	–	µs
t_h	Hold Time		1	.5	–	µs
t_{clk}	Clock Pulse Width		0.5	1	–	µs
t_{pwv}	Verify Pulse Width		20	30	–	µs
t_{pwp}	Programming Pulse Width		40	–	100	ms
t_{bew}	Bulk Erase Pulse Width		200	–	–	ms
t_{rst}	Reset Time From Valid V_{CCP}		45	–	–	µs

Figure 19. Timing Waveform for isp Operation

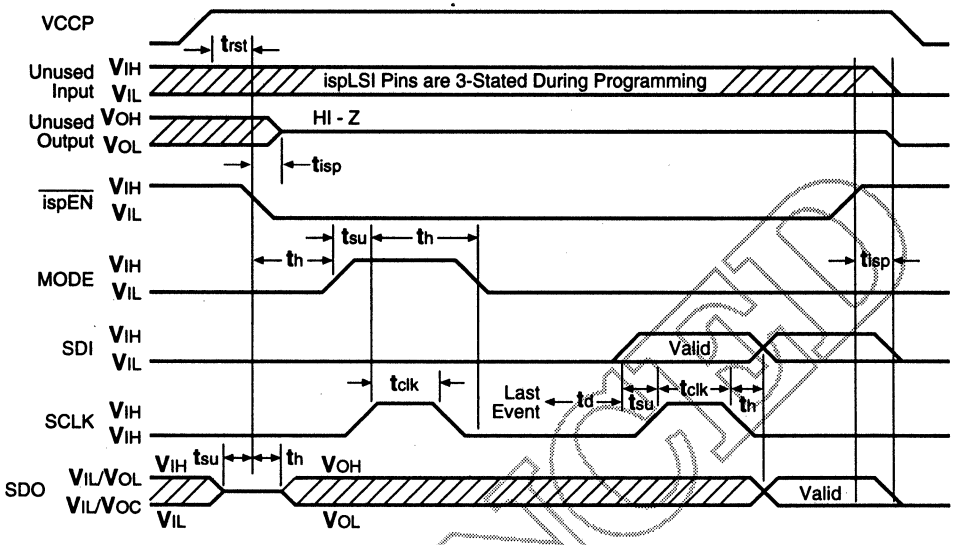
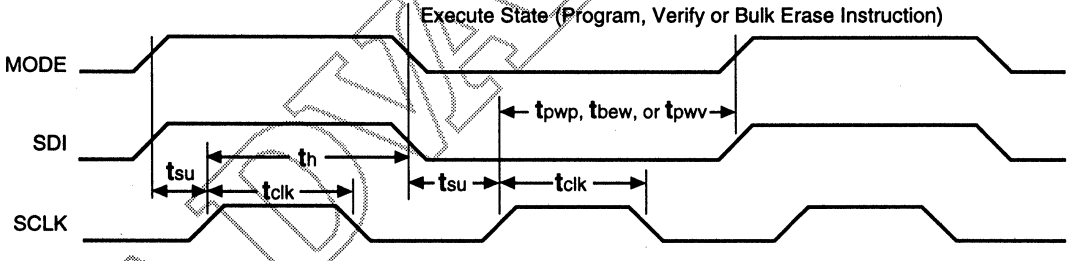
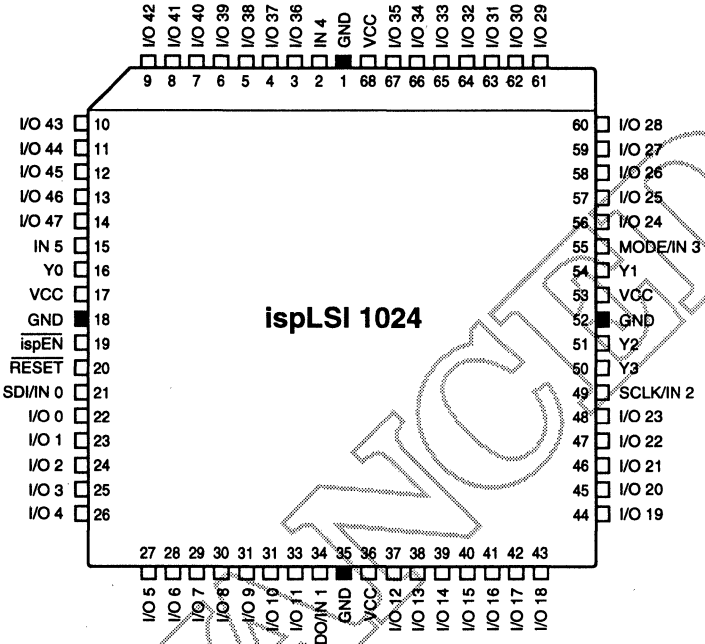


Figure 20. Program, Verify & Bulk Erase Waveform

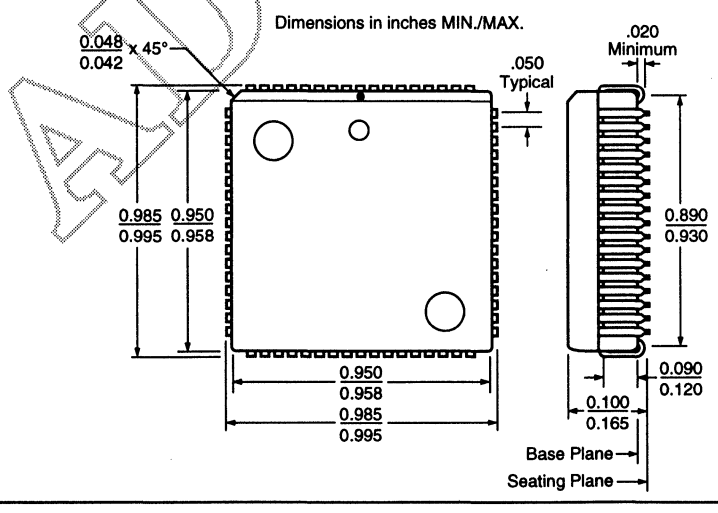


Pin Configuration

ispLSI 1024 PLCC Pinout Diagram



68-Pin PLCC



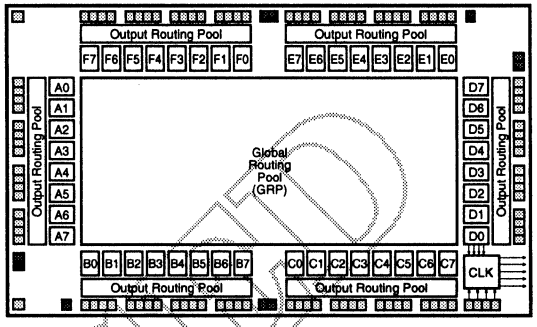
ispLSI™ 1048

in-system programmable Large Scale Integration

Features

- **in-system programmable HIGH DENSITY LOGIC**
  - **— Member of Lattice's ispLSI Family**
  - **— Fully Compatible with Lattice's pLSI™ Family**
  - **— High Speed Global Interconnects**
  - **— 96 I/O Pins, Ten Dedicated Inputs**
  - **— 288 Registers**
  - **— Wide Input Gating for Fast Counters, State Machines, Address Decoders, etc.**
  - **— Small Logic Block Size for Random Logic**
  - **— Security Cell Prevents Unauthorized Copying**
- **HIGH PERFORMANCE E²CMOS® TECHNOLOGY**
  - **— f_{max} = 70 MHz Maximum Operating Frequency**
  - **— t_{pd} = 20 ns Propagation Delay**
  - **— TTL Compatible Inputs and Outputs**
- **in-system programmable 5-VOLT ONLY**
  - **— Change Logic and Interconnects "on-the-fly" in Seconds**
  - **— Reprogram Soldered Device for Debugging**
  - **— Non-Volatile E²CMOS Technology**
  - **— 100% Tested**
- **COMBINES EASE OF USE AND THE FAST SYSTEM SPEED OF PLDs WITH THE DENSITY AND FLEXIBILITY OF FIELD PROGRAMMABLE GATE ARRAYS**
  - **— Complete Programmable Device can Combine Glue Logic and Structured Designs**
  - **— 100% Routable at 80% Utilization**
  - **— Four Dedicated Clock Input Pins**
  - **— Synchronous and Asynchronous Clocks**
  - **— Flexible Pin Placement**
  - **— Optimized Global Routing Pool Allows Global Interconnectivity**
- **pLSI/ispLSI DEVELOPMENT SYSTEM (pDS™)**
  - **— Boolean Logic Compiler**
  - **— Automatic Place and Route**
  - **— Manual Partitioning**
  - **— PC Platform**
  - **— Easy to Use Windows Interface**
- **ADVANCED pLSI/ispLSI DEVELOPMENT SYSTEM**
  - **— Industry Standard, Third Party Design Environments**
  - **— Schematic Capture**
  - **— Fully Automatic Partitioning**
  - **— Automatic Place and Route**
  - **— Comprehensive Logic and Timing Simulation**
  - **— PC and Workstation Platforms**

Functional Block Diagram



Description

The Lattice ispLSI 1048 is a High Density Programmable Logic Device featuring 5-Volt in-system programmability and in-system diagnostic capabilities. The device contains 288 Registers, 96 Universal I/O pins, ten Dedicated Input Pins, four Dedicated Clock Input Pins and a Global Routing Pool (GRP). The GRP provides complete interconnectivity between all of these elements. It is the first device which offers non-volatile "on-the-fly" reprogrammability of the logic, as well as the interconnects to provide truly reconfigurable systems. It is architecturally and parametrically compatible to the pLSI 1048 device, but multiplexes four of the dedicated input pins to control in-system programming.

The basic unit of logic on the ispLSI 1048 device is the Generic Logic Block (GLB). The GLBs are labeled A0, A1 .. F7 (see figure 1). There are a total of 48 GLBs in the ispLSI 1048 device. Each GLB has 18 inputs, a programmable AND/OR/Exclusive OR array, and four outputs which can be configured to be either combinatorial or registered. Inputs to the GLB come from the GRP. All of the GLB outputs are brought back into the GRP so that they can be connected to the inputs of any other GLB on the device.

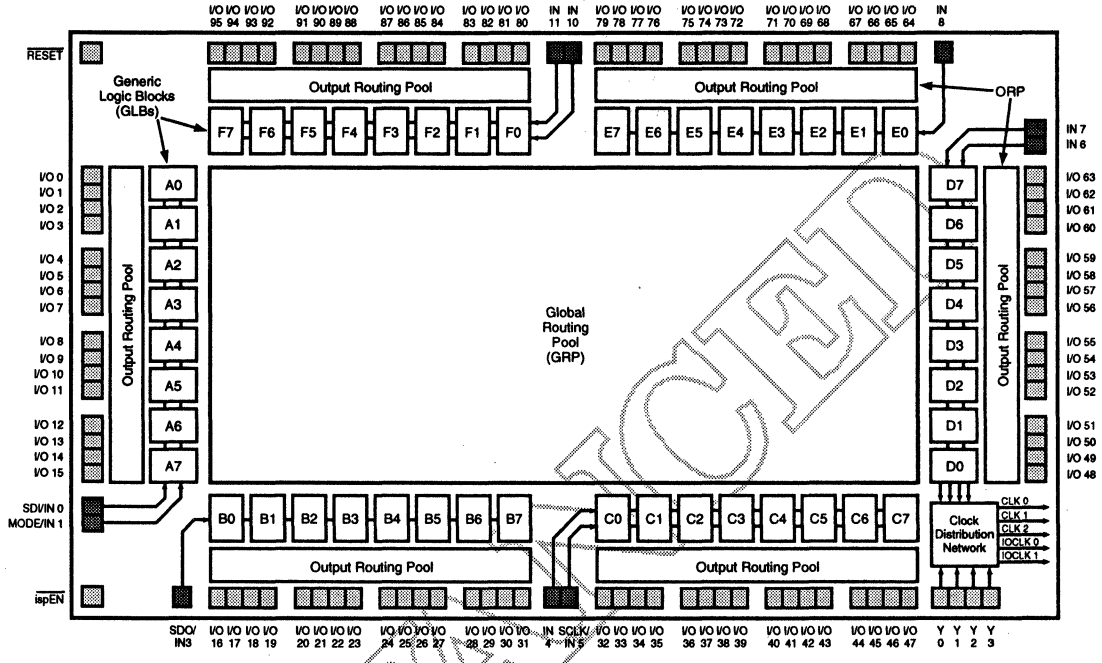
The device also has 96 I/O Cells, each of which is directly connected to an I/O pin. Each I/O cell can be individually programmed to be a combinatorial input, registered input, latched input, output or bi-directional I/O pin with 3-state. The signal levels are TTL compatible voltages and the output drivers can source 4 mA or sink 8 mA.

LATTICE SEMICONDUCTOR CORP., 5555 Northeast Moore Ct., Hillsboro, Oregon 97124, U.S.A.
Tel. 1-800-LATTICE (528-8423); FAX (503) 681-3037

January 1992. Rev. A

Functional Block Diagram

Figure 1. ispLSI 1048



Description (continued)

The 96 I/O Cells are grouped into three sets of 16 each. as shown in figure 1. Each of these I/O groups is associated with a logic Megablock through the use of the Output Routing Pool (ORP) and shares a common Output Enable (OE) signal.

Eight GLBs, 16 I/O Cells and one ORP are connected together to make a logic Megablock. The Megablock is defined by the resources that it shares. The outputs of the eight GLBs are connected to a set of 16 universal I/O cells by the ORP. The ispLSI 1048 Device contains six of these Megablocks.

The GRP has as its inputs the outputs from all of the GLBs and all of the inputs from the bi-directional I/O cells. All of these signals are made available to the inputs of the GLBs. Delays through the GRP have been equalized to minimize timing skew and logic glitching.

Clocks in the ispLSI 1048 device are selected using the Clock Distribution Network. Four dedicated clock pins (Y0 to Y3) are brought into the distribution network, and five outputs (CLK 0 to CLK 2 and IOCLK 0, IOCLK 1) are provided to route clocks to the GLBs and I/O cells. The Clock Distribution Network can also be driven from a special GLB (D0 on the ispLSI 1048 device). The logic of this GLB allows the user to create an internal clock from a combination of internal signals within the device.

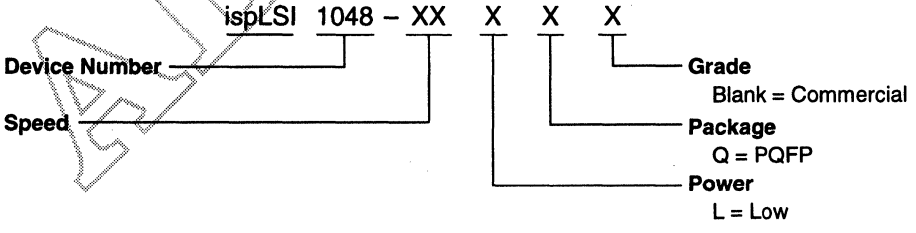
The ispLSI 1048 device is part of Lattice's in-system programmable Large Scale Integration (ispLSI) family. This family contains a range of devices from the ispLSI 1016, with 96 registers, to the ispLSI 1048 with 288 registers. The ispLSI Family Product Selector Guide below lists key attributes of the devices along with the number of resources available.

2

ispLSI Family Product Selector Guide

DEVICE	ispLSI 1016	ispLSI 1024	ispLSI 1032	ispLSI 1048
GLBs	16	24	32	48
Registers	96	144	192	288
I/O Pins	32	48	64	96
Dedicated Inputs	4	6	8	10
Pin Count	44	68	84	120

Ordering Information



Pin Description

Name	PLCC Pin Numbers	Description
I/O 0 - I/O 5 I/O 6 - I/O 11 I/O 12 - I/O 17 I/O 18 - I/O 23 I/O 24 - I/O 29 I/O 30 - I/O 35 I/O 36 - I/O 41 I/O 42 - I/O 47 I/O 48 - I/O 53 I/O 54 - I/O 59 I/O 60 - I/O 65 I/O 66 - I/O 71 I/O 72 - I/O 77 I/O 78 - I/O 83 I/O 84 - I/O 89 I/O 90 - I/O 95	20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99,100,101,102,103, 109,110,111,112,113,114, 115,116,117,118,119,120, 1, 2, 3, 4 5, 6, 7, 8, 9, 10, 11, 12	Input/Output Pins - These are the general purpose I/O pins used by the logic array.
IN 4 IN 6 - IN 11	48, 79,104,105, – 108, 13	Dedicated input pins to the device. (IN 9 not available)
$\overline{\text{ispEN}}$	17	Input – Dedicated in-system programming enable input pin. This pin is brought low to enable the programming mode. The MODE, SDI, SDO and SCLK options become active.
SDI/IN 0	19	Input – This pin performs two functions. It is a dedicated input pin when ispEN is logic high. When ispEN is logic low, it functions as an input pin to load programming data into the device. SDI/IN 0 also is used as one of the two control pins for the isp state machine.
MODE/IN 1	44	Input – This pin performs two functions. It is a dedicated input pin when $\overline{\text{ispEN}}$ is logic high. When $\overline{\text{ispEN}}$ is logic low, it functions as a pin to control the operation of the isp state machine.
SDO/IN 3	47	Input/Output – This pin performs two functions. It is a dedicated clock input pin when $\overline{\text{ispEN}}$ is logic high. When $\overline{\text{ispEN}}$ is logic low, it functions as an output pin to read serial shift register data. Input – This pin performs two functions. It is a dedicated input when $\overline{\text{ispEN}}$ is logic high. When $\overline{\text{ispEN}}$ is logic low, it functions as a clock pin for the Serial Shift Register.
$\overline{\text{RESET}}$	18	Active Low (0) Reset pin which resets all of the GLB and I/O registers in the device.
Y0	14	Dedicated Clock input. This clock input is connected to one of the clock inputs of all of the GLBs on the device.
Y1	78	Dedicated clock input. This clock input is brought into the clock distribution network, and can optionally be routed to any GLB on the device.
Y2	75	Dedicated clock input. This clock input is brought into the clock distribution network, and can optionally be routed to any GLB and/or any I/O Cell on the device.
Y3	74	Dedicated clock input. This clock input is brought into the clock distribution network, and can optionally be routed to any I/O Cell on the device.
GND VCC	46, 76,106, 16 15, 45, 77, 107	Ground (GND) V_{CC}

Absolute Maximum Ratings [1]

Supply Voltage V_{CC} . . . . . -0.5 to +7.0V

Input Voltage Applied. . . . . .-2.5 to V_{CC} +1.0V

Off-State Output Voltage Applied . . . . . .-2.5 to V_{CC} +1.0V

Storage Temperature . . . . . .-65 to 125°C

Ambient Temp. with Power Applied . . . . . .-55 to 125°C

1. Stresses above those listed under the "Absolute Maximum Ratings" may cause permanent damage to the device. Functional operation of the device at these or at any other conditions above those indicated in the operational sections of this specification is not implied (while programming, follow the programming specifications).

DC Recommended Operating Condition

SYMBOL	PARAMETER	MIN.	MAX.	UNITS
T_A	Ambient Temperature	0	70	°C
V_{CC}	Supply Voltage	4.75	5.25	V
V_{IL}	Input Low Voltage	0	0.8	V
V_{IH}	Input High Voltage	2.0	V_{CC}	V

Capacitance (T_A=25°C,f=1.0 MHz)

SYMBOL	PARAMETER	MAXIMUM[1]	UNITS	TEST CONDITIONS
C_1	Input Capacitance	8	pf	V_{CC}=5.0V, V_{IN}=2.0V
C_2	I/O, Y Capacitance	10	pf	V_{CC}=5.0V, VI/O, Y=2.0V

1. Guaranteed but not 100% tested.

Data Retention Specifications

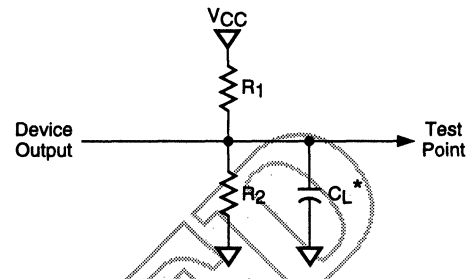
PARAMETER	MINIMUM	MAXIMUM	UNITS
Data Retention	20	–	YEARS
Erase/Reprogram Cycles	–	1000	CYCLES

Switching Test Conditions

Input Pulse Levels	GND to 3.0V
Input Rise and Fall Times	3ns 10% to 90%
Input Timing Reference Levels	1.5V
Output Timing Reference Levels	1.5V
Output Load	See Figure 2

3-state levels are measured 0.5V from steady-state active level.

Figure 2. Test Load



Output Load Conditions (see figure 2)

Test Condition		R1	R2	CL
1		470Ω	390Ω	35pF
2	Active High	∞	390Ω	35pF
	Active Low	470Ω	390Ω	35pF
3	Active High to Z at V_{OH} - 0.5V	∞	390Ω	5pF
	Active Low to Z at V_{OL} + 0.5V	470Ω	390Ω	5pF

DC Electrical Characteristics

Over Recommended Operating Conditions

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNITS
V_{OL}	Output Low Voltage	I_{OL} =8 mA.	–	–	0.4	V
V_{OH}	Output High Voltage	I_{OH} =-4 mA.	2.4	–	–	V
I_{IL}	Input or I/O Low Leakage Current	$0V \le V_{IN} \le V_{IL}$ (MAX.)	–	–	-10	µA
I_{IH}	Input or I/O High Leakage Current	$V_{IH} \le V_{IN} \le V_{CC}$	–	–	10	µA
I_{OS}[1]	Output Short Circuit Current	V_{CC} = 5V, V_{OUT} = 0.5V	-60	–	-200	mA
I_{CC}[2]	Operating Power Supply Current	V_{IL} = 0.5V, V_{IH} = 3.0V F_{TOGGLE} = 20 MHz	–	–	–	mA

1. One output at a time for a maximum duration of one second. (Vout = 0.5V)
2. Measured at a frequency of 20 MHz using twelve 16-bit counters.

External Switching Characteristics[1, 2, 3] ispLSI 1048-70

Over Recommended Operating Conditions

PARAMETER	TEST[6] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
tpd1	1	1	Data Propagation Delay, 4PT bypass, ORP bypass	–	–	20	ns
tpd2	1	2	Data Propagation Delay	–	–	25	ns
tco1[5]	1	3	External Clock to Output Delay, ORP bypass	–	–	–	ns
tco2[5]	1	4	External Clock to Output Delay	–	–	–	ns
tco3	1	5	Internal Synch. Clock to Output Delay	–	–	–	ns
tco4	1	6	Asynchronous Clock to Output Delay	–	–	–	ns
tr1	1	7	External Pin Reset to Output Delay	–	–	–	ns
tr2	1	8	Asynchronous PT Reset to Output Delay	–	–	–	ns
ten	2	9	Input to Output Enable	–	–	–	ns
tdis	3	10	Input to Output Disable	–	–	–	ns

External AC Recommended Operating Conditions[1, 2, 3] ispLSI 1048-70

Over Recommended Operating Conditions

PARAMETER	TEST[6] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
fmax[4]	1	11	Clock Frequency with Internal Feedback	–	–	70	MHz
fmax (External)	1	12	Clock Frequency with External Feedback	–	–	–	MHz
tsu1	–	13	Setup Time before External Synch. Clock, 4PT bypass	–	–	–	ns
tsu2	–	14	Setup Time before External Synch Clock	–	–	–	ns
tsu3	–	15	Setup Time before Internal Synch. Clock	–	–	–	ns
tsu4	–	16	Setup Time before Asynchronous Clock	–	–	–	ns
th1	–	17	Hold time after External Synchronous Clock, 4PT bypass	–	–	–	ns
th2	–	18	Hold time after External Synchronous Clock	–	–	–	ns
th3	–	19	Hold time after Internal Synchronous Clock	–	–	–	ns
th4	–	20	Hold time after Asynchronous Clock	–	–	–	ns
trw1	–	21	External Reset Pulse Duration	–	–	–	ns
trw2	–	22	Asynchronous Reset Pulse Duration	–	–	–	ns
twh1, twl1	–	23,24	External Synchronous Clock Pulse Duration, High, Low	–	–	–	ns
twh2, twl2	–	25,26	Asynchronous Clock Pulse Duration, High, Low	–	–	–	ns

1. External Switching Characteristics are tested and guaranteed.
2. See Timing Technical Note for further details.
3. Unless noted, all parameters use ORP, GRP fanout of four, PTSA, and are measured with 16 outputs switching.
4. Standard 16-bit counter implementation using GRP feedback.
5. Clock to output specifications include a maximum skew of 2 ns.
6. Refer to Switching Test Conditions section.

Switching Characteristics[1, 2, 3] — ispLSI 1048-70

Using I/O Cell

PARAMETER	TEST[4] COND.	#	DESCRIPTION	MIN.	TYP.	MAX.	UNITS
t_{su5}	–	27	Setup Time before External Synchronous Clock	–	–	–	ns
t_{su6}	–	28	Setup Time before Internal Synchronous Clock	–	–	–	ns
t_{h5}	–	29	Hold Time after External Synchronous Clock	–	–	–	ns
t_{h6}	–	30	Hold Time after Internal Synchronous Clock	–	–	–	ns
t_{wh3}, t_{wl3}	–	31,32	Clock Pulse Duration, High, Low	–	–	–	ns

1. External Parameters are tested and guaranteed.
2. See Timing Technical Note for further details.
3. Unless noted, all parameters use ORP, GRP fanout of four, PTSA, and are measured with 16 outputs switching.
4. Refer to Switching Test Conditions section.

Architectural Description

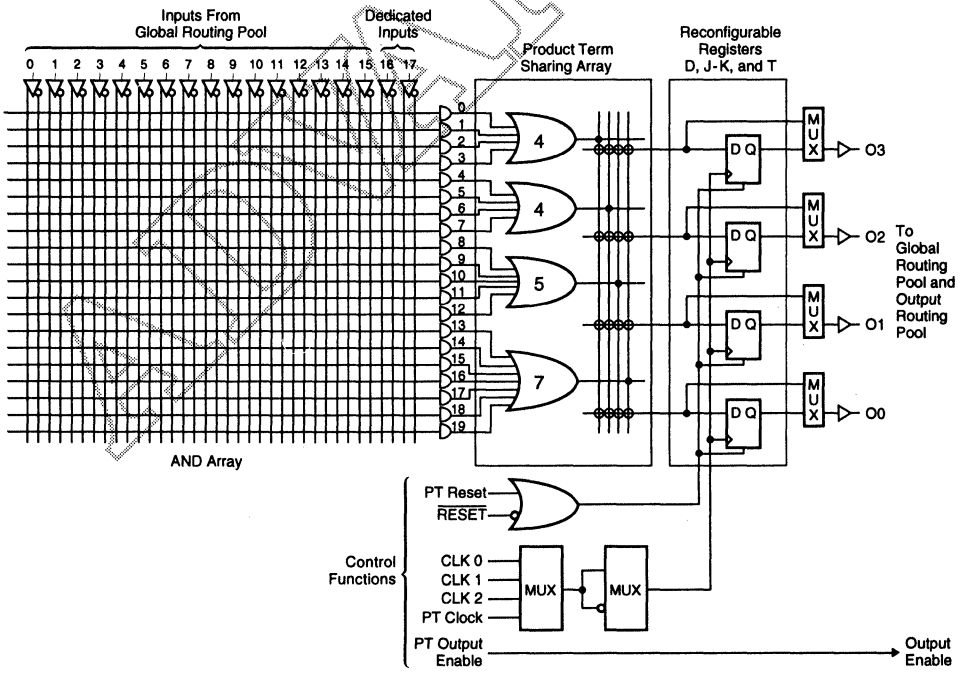
The Generic Logic Block

The Generic Logic Block (GLB) is the standard logic block of the Lattice High Density ispLSI Device. This GLB has 18 inputs, four outputs and the logic necessary to implement most standard logic functions. The internal logic of the GLB is divided into four separate sections (see figure 3). The AND Array, the Product Term Sharing Array (PTSA), the Reconfigurable Registers, and the Control Functions. The AND array consists of 20 product terms which can produce the logical sum of any of the 18 GLB inputs. Sixteen of the inputs come from the Global Routing Pool, and are either feedback signals from any of the 48 GLBs or inputs from the external I/O Cells. The two remaining inputs come directly from two dedicated input pins. These signals are available to the product terms in both the logical true and the complemented forms which makes boolean logic reduction easier.

The PTSA takes the 20 product terms and allocates them to the four GLB outputs. There are four OR gates, with four, four, five and seven product terms (see figure 3). The output of any of these gates can be routed to any of the four GLB outputs, and if more product terms are needed, the PTSA can combine them as necessary. If the users' main concern is speed, the PTSA can use a bypass circuit with four product terms to increase the performance of the cell (see figure 4). This can be done to any or all of the four outputs from the GLB.

The Reconfigurable Registers consist of four D-type flip-flops with an Exclusive OR Gate on the input. The Exclusive OR gate in the GLB can be used either as a logic element or to reconfigure the D-type flip-flop to emulate a J-K, or T-type flip-flop (see figure 5). This greatly simplifies the design of counters, comparators and ALU type functions. The registers can be bypassed if the user needs a combinatorial output. Each register output is brought back into the Global Routing Pool and is also brought to the I/O cells via the Output Routing Pool. Reconfigurable registers are not available when the four product term bypass is used.

Figure 3. GLB: Product Term Sharing Array



Architectural Description

Figure 4. GLB: Four Product Term Bypass

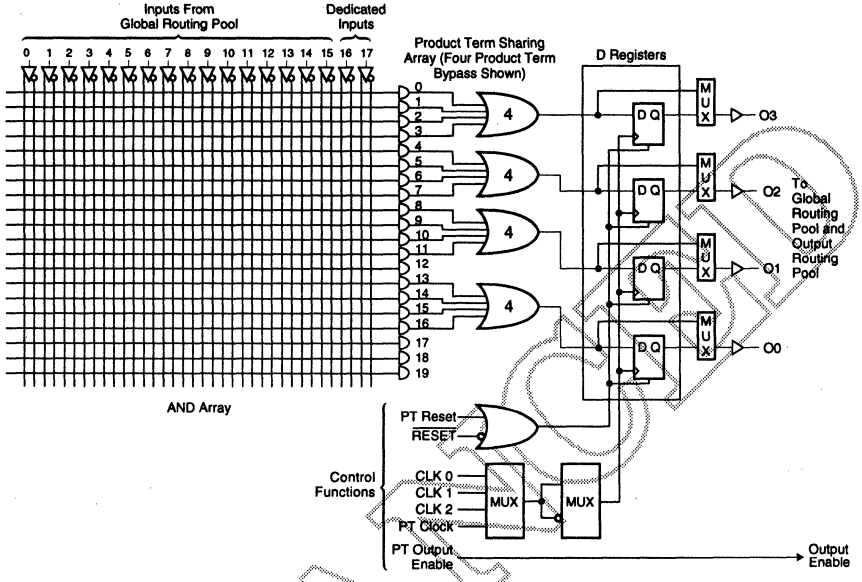
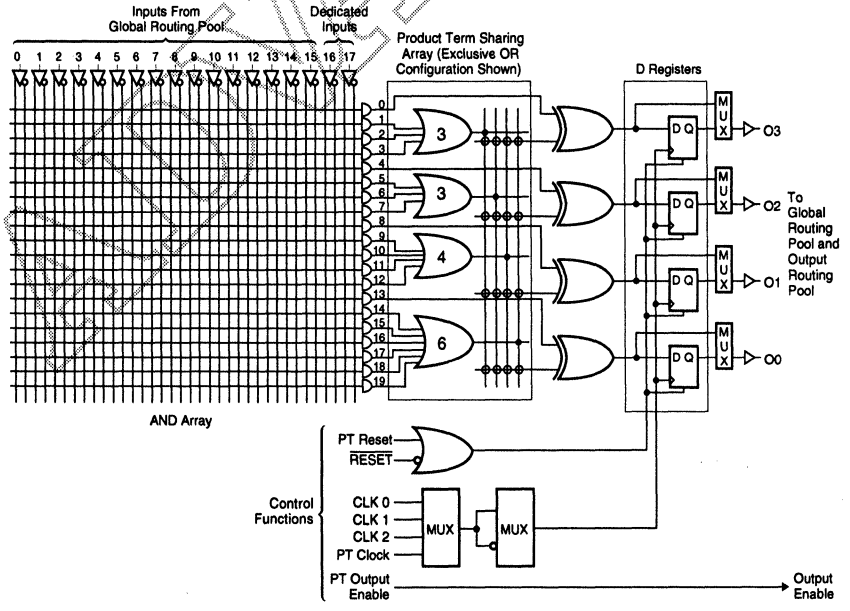


Figure 5. GLB: Exclusive OR Gate



Architectural Description

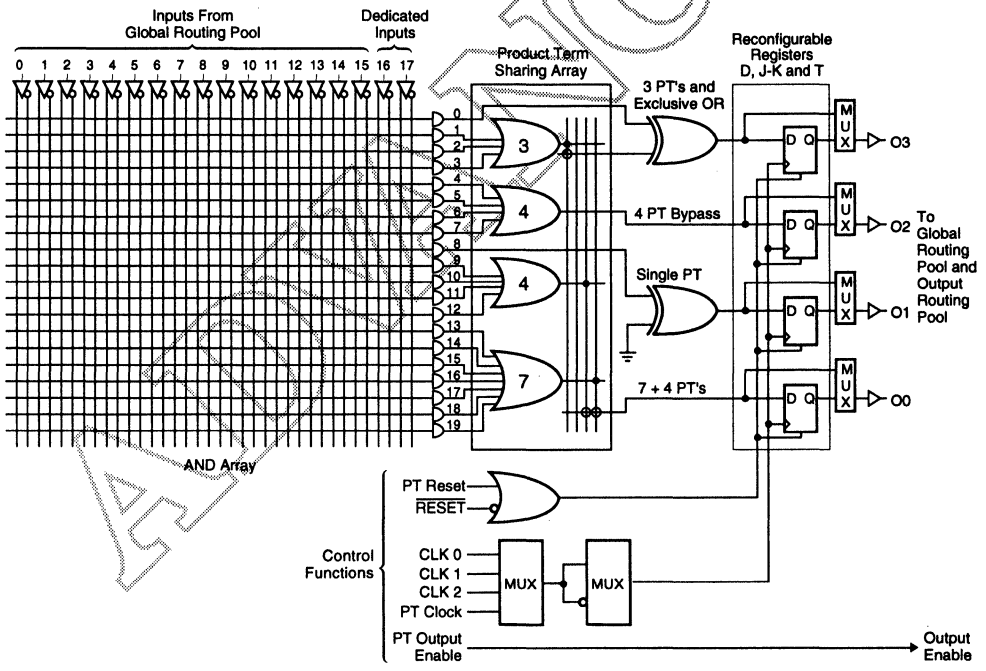
The Generic Logic Block (continued)

The PTSA is flexible enough to allow these features to be used in virtually any combination that the user desires. In the GLB shown in figure 6, Output Three (O3) is configured using the XOR Gate and Output Two (O2) is configured using the four Product Term Bypass. Output One (O1) uses one of the inputs from the five Product Term OR gate while Output Zero (O0) combines the remaining four product terms with all of the product terms from the seven Product Term OR gate for a total of eleven (7+4).

Various signals which control the operation of the GLB are developed in the Control Function section. The clock for the registers can come from any of three sources developed in the Clock Distribution Network (See Clock Distribution Network Section) or from a product term within the GLB. The Reset Signal for the GLB can come from the Global Reset pin or from a product term within the block. The Output Enable for the I/O cells associated with the GLB comes from a product term within the block. Use of a product term for a control function makes that product term unavailable for use as a logic term. Refer to the following table to determine which logic functions are affected.

There are additional features in a GLB which allow implementation of logic intensive functions. These features are accessible using the hard Macros from the software and require no intervention on the part of the user.

Figure 6. GLB: Various Logical Combinations



Architectural Description

Product Term Sharing Matrix

Product Term #	Standard Configuration Output Number 3 2 1 0	Four Product Term Bypass Output Number 3 2 1 0	Single Product Term Output Number 3 2 1 0	XOR Function Output Number 3 3 2 2 1 1 0 0	Alternate Function
0	■ ■ ■ ■	■	■	■	
1	■ ■ ■ ■	■		■	
2	■ ■ ■ ■	■		■	
3	■ ■ ■ ■	■		■	
4	■ ■ ■ ■	■	■	■	
5	■ ■ ■ ■	■		■	
6	■ ■ ■ ■	■		■	
7	■ ■ ■ ■	■		■	
8	■ ■ ■ ■	■	■	■	
9	■ ■ ■ ■	■		■	
10	■ ■ ■ ■	■		■	
11	■ ■ ■ ■	■		■	
12	■ ■ ■ ■	■		■	■ CLK/Reset
13	■ ■ ■ ■	■	■	■	
14	■ ■ ■ ■	■		■	
15	■ ■ ■ ■	■		■	
16	■ ■ ■ ■	■		■	
17	■ ■ ■ ■			■	
18	■ ■ ■ ■			■	
19	■ ■ ■ ■			■	■ OE/Reset

This matrix shows how each of the product terms are used in the various modes. As an example, Product Term 12 can be used as an input to the five input OR gate in the standard configuration. This OR gate under standard configuration can be routed to any of the four GLB outputs. Product Term 12 is not used in the four product term bypass mode. When GLB output one is used in the Exclusive OR (XOR) mode Product Term 12 becomes one of the inputs to the four input OR Gate. If Product Term 12 is not used in the logic, then it is available for use as either the Asynchronous Clock signal or the GLB Reset signal.

Architectural Description

The Megablock

The ispLSI family is structured into multiple "Megablocks". A Megablock consists of 8 GLBs, an Output Routing Pool (ORP) and 16 I/O Cells. Each of these will be explained in detail in the following sections. These elements are coupled together as shown in figure 7. This logic structure is referred to as the Megablock. The various members of the ispLSI family are created by combining from two to six Megablocks on a single device.

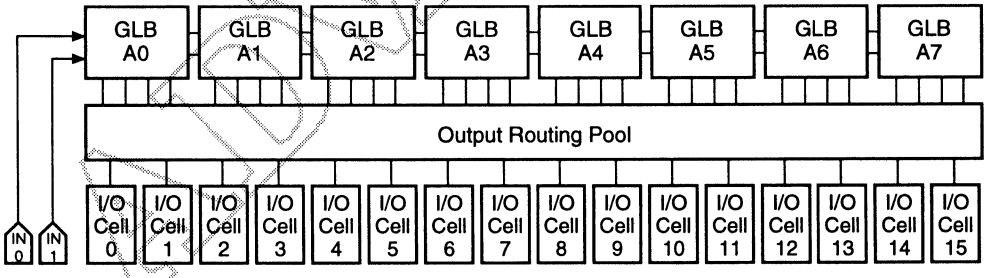
The Megablock shares two sets of resources. The eight GLBs within the Megablock share two dedicated input pins. These dedicated input pins are not available to GLBs in any other Megablock. These pins are dedicated (non-registered) inputs only and are automatically assigned by software. One Output Enable signal is generated within the Megablock and is common to all sixteen of the I/O Cells in the Megablock. The Output Enable signal can be generated using a product term in any of the eight GLBs within the Megablock (See the following section on the Output Enable Multiplexers).

Because of the shared logic within the Megablock, signals which share a common function (counters, busses, etc.) should be grouped within a Megablock. This will allow the user to obtain the best utilization of the logic within the device and eliminate routing bottlenecks.

Megablocks in Each Device

DEVICE	MEGABLOCKS	GLBs	I/O CELLS
ispLSI 1016	2	16	32
ispLSI 1024	3	24	48
ispLSI 1032	4	32	64
ispLSI 1048	6	48	96

Figure 7. The Megablock



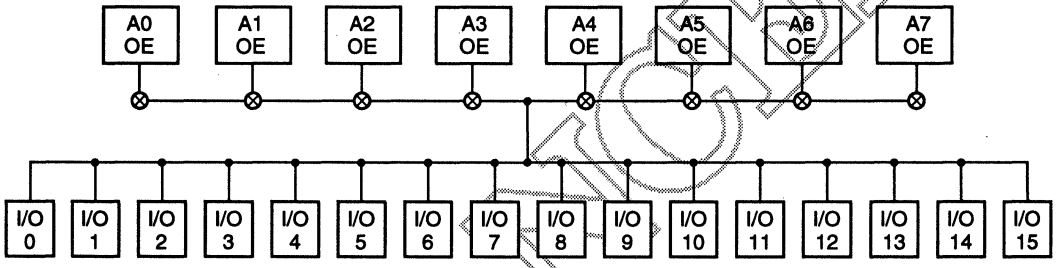
Note: The inputs from the I/O Cell are not shown in this figure, they go directly to the GRP.

Architectural Description

The Output Enable Control

One Output Enable (OE) signal can be generated within each GLB using the OE Product Term 19. One of the eight OE signals from each GLB within a Megablock, is then routed to all of the I/O Cells within the Megablock (see figure 8). This OE signal can simultaneously control all of the 16 I/O cells which are used in 3-state mode. Individual I/O cells also have independent control for permanently enabling or disabling the output buffer (Refer to the I/O Cell Section). Only one OE signal is allowed per Megablock for 3-state operation. The advantage to this approach is that the OE signal can be generated in any GLB within the Megablock which happens to have an unused OE product term. This frees up the other OE product terms for use as logic.

Figure 8. Output Enable Controls



Architectural Description

The Output Routing Pool

The Output Routing Pool (ORP) routes signals from the Generic Logic Block outputs to I/O Cells configured as outputs or bi-directional pins (see figure 9). The purpose of the ORP is to allow greater flexibility when assigning I/O pins. It also simplifies the job for the routing software which results in a higher degree of utilization.

By examining the programmable switch matrix in figure 9, it can be seen that a GLB output can be connected to one of four I/O Cells. Further flexibility is provided by using the PTSA, (Figures 3 through 7) which makes the GLB outputs completely interchangeable. This allows the routing program to freely interchange the outputs to achieve the best routability. This is an automatic process and requires no intervention on the part of the user.

The ORP bypass connections (see figure 10) further increase the flexibility of the device. The ORP bypass connect specific GLB outputs to specific I/O Cells at a faster speed. The bypass path tends to restrict the routability of the device and should only be used for critical signals.

Figure 9. Output Routing Pool

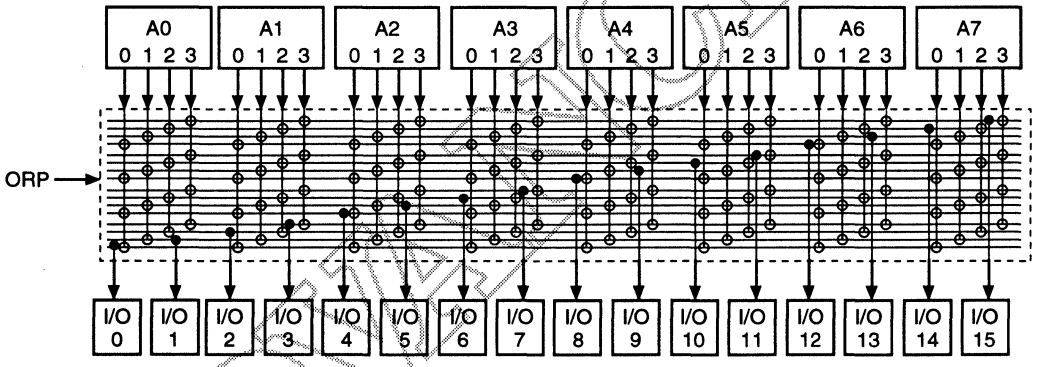
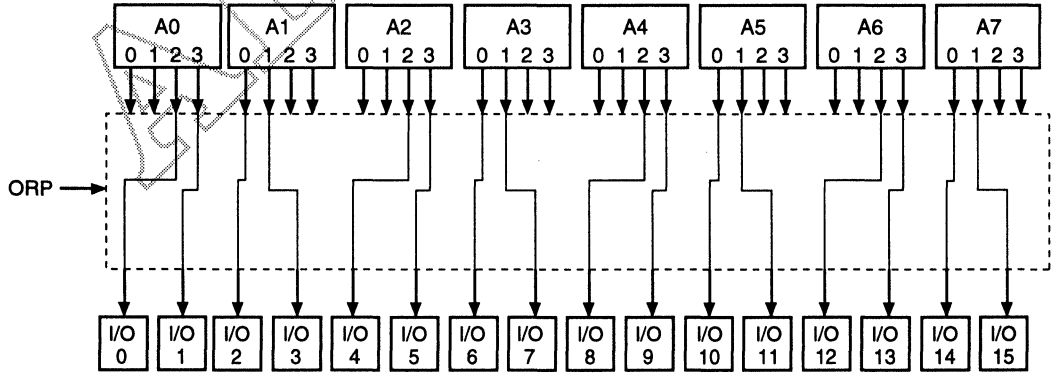


Figure 10. Output Routing Pool showing Bypass



Architectural Description

I/O Cell

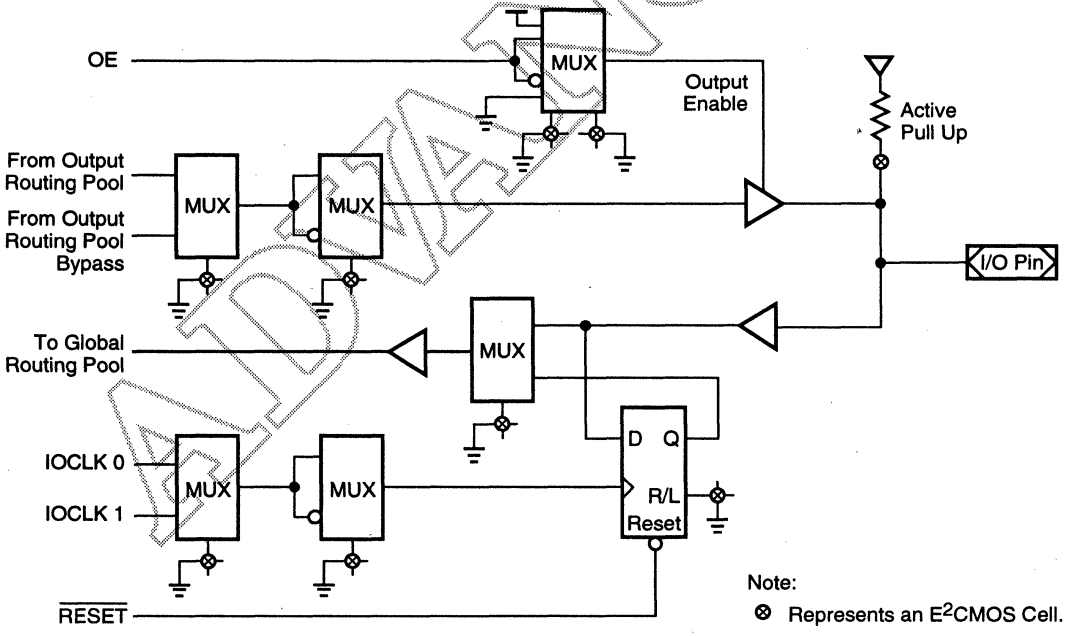
The I/O Cell (see figure 11) is used to route input, output or bidirectional signals connected to the I/O pin. The two logic inputs come from the ORP (see figure 7). One comes from the ORP, and the other comes from the faster ORP bypass. A pair of multiplexers select which signal will be used, and its polarity.

The OE signal comes from the Output Enable. As with the data path, a multiplexer selects the signal polarity. The Output Enable can be set to a logic high (Enabled) when a straight output pin is desired, or logic low (Disabled) when a straight input pin is needed. The Global Reset ($\overline{\text{RESET}}$) signal is driven by the active low chip reset pin. Each I/O Cell can individually select one of the two clock signals (IOCLK 0 or IOCLK 1). These clock signals are generated by the Clock Distribution Network.

Using the multiplexers, the I/O Cell can be configured as an input, an output, a 3-stated output or a bidirectional I/O. The D-type register can be configured as a level sensitive transparent latch or an edge triggered flip-flop to store the in-coming data. Figure 12 illustrates some of the various I/O cell configurations possible.

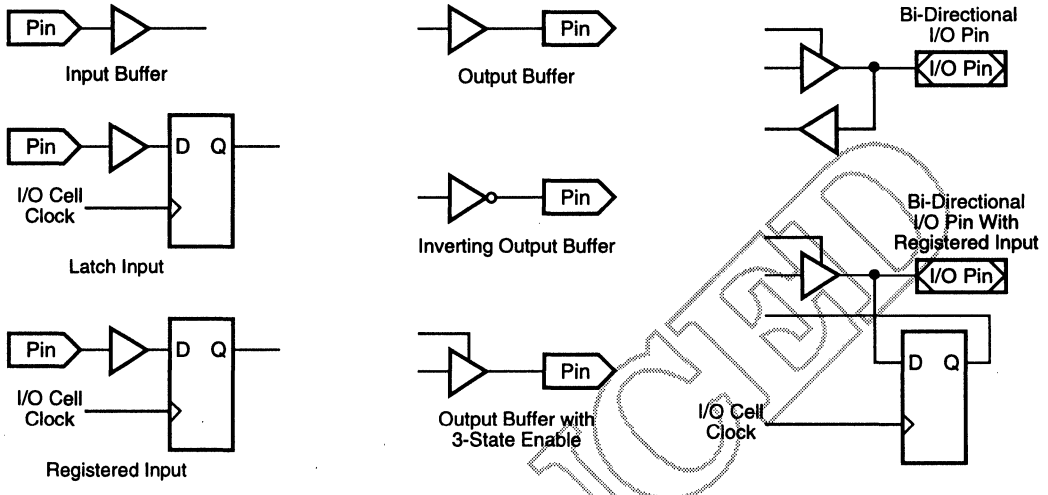
There is an Active Pullup resistor on the I/O pins which is automatically used when the pin is not connected. This prevents the pin from floating and inducing noise into the device or consuming additional power.

Figure 11. I/O Cell Architecture



Architectural Description

Figure 12. Example I/O Cell Configurations



Architectural Description

Clock Distribution Network

The Clock Distribution Network is shown in figure 13. It generates five global clock signals CLK 0, CLK 1, CLK 2 and IOCLK 0, IOCLK 1. The first three, CLK 0, CLK 1 and CLK 2 are used for clocking all the GLBs in the device. Similarly, IOCLK 0 and IOCLK 1 signals are used for clocking all of the I/O Cells in the device. There are four dedicated system clock pins (Y0, Y1, Y2, Y3) which can be directed to any GLB or any I/O Cell using the Clock Distribution Network. The other inputs to the Clock Distribution Network are the 4 outputs of a dedicated clock GLB ("D0" for ispLSI 1048). These Clock GLB outputs can be used to create a user-defined internal clocking scheme.

Typically the clock GLB will be clocked using an external main clock pin Y0 connected to global clock signal CLK 0. The outputs of the clock GLB in turn will generate "divide by two" or "divide by four" phases of the CLK 0 which can be connected to CLK 1, CLK 2 or IOCLK 0, IOCLK 1 global clocks.

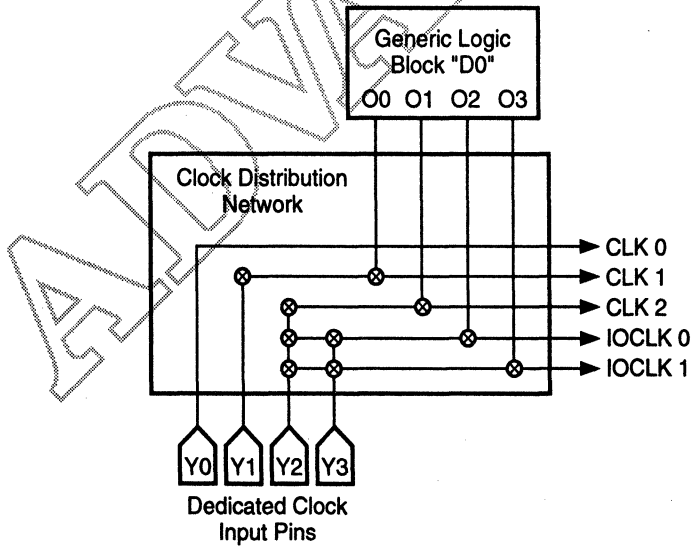
All GLBs also have the capability of generating their own asynchronous clocks using Product Term 12. CLK 0, CLK 1 and CLK 2 feed to their corresponding inputs on all the GLBs (see figure 3).

The two I/O clocks generated in the Clock Distribution Network IOCLK 0 and IOCLK 1, are brought to all 96 of the I/O cells and the user programs the I/O cell to use one of the two.

Global Routing Pool

The Global Routing Pool (GRP) is a Lattice proprietary interconnect structure which offers fast predictable speeds with complete connectivity. The GRP allows the outputs from the GLBs or the I/O cell inputs to be connected to the inputs of the GLBs. Any GLB output is available to the input of all other GLBs, and similarly an input from an I/O pin is available as an input to all of the GLBs. Because of the uniform architecture of the ispLSI device, the delays through the Global Routing Pool are both consistent and predictable. However, they are slightly affected by fanout.

Figure 13. Clock Distribution Network



Note: Y3 pin should always be used first as an IOCLK 0 or IOCLK 1 before using Y2 pin.

Security Cell

A security cell is provided in the ispLSI devices to prevent unauthorized copying of the array patterns. Once programmed, this cell prevents further read access to the functional bits in the device. This cell can only be erased by reprogramming the device, so the original configuration can never be examined once this cell is programmed.

Device Programming

ispLSI devices are programmed using a Lattice-approved Device Programmer, available from a number of third party manufacturers or in-system using Lattice programming algorithms. Complete programming of the device takes only a few seconds. Erasing of the device is automatic and is completely transparent to the user.

Latch-up Protection

ispLSI devices are designed with an on-board charge pump to negatively bias the substrate. The negative bias is of sufficient magnitude to prevent input undershoots from causing the internal circuitry to latchup. Additionally, outputs are designed with n-channel pull-ups instead of the traditional p-channel pull-ups to eliminate any possibility of SCR induced latching.

In-System Programmability

The ispLSI devices are the in-system programmable version of the Lattice high density programmable Large Scale Integration (pLSI) devices. By integrating all the high voltage programming circuitry on-chip, the programming can be accomplished by simply shifting data into the device. Once the function is programmed, the non-volatile E^2CMOS cells will not lose the pattern even when the power is turned off.

All necessary programming is done via five TTL level logic interface signals. These five signals are fed into the on-chip programming circuitry where a state machine controls the programming. The simple signals for interface include isp Enable ($\overline{ispEN}$), Serial Data In (SDI), Serial Data Out (SDO), Serial Clock (SCLK) and Mode (MODE) control. Figure 17 illustrates the block diagram of one possible scheme of the programming interface for the isp devices. For details on the operation of the internal state machine and programming of the device please refer to the In-system programming application note.

Figure 17. isp Programming Interface

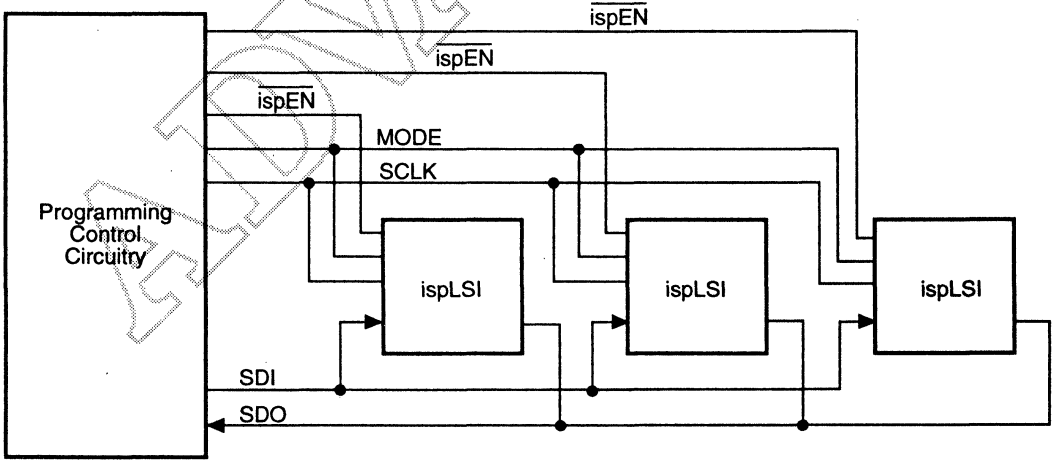
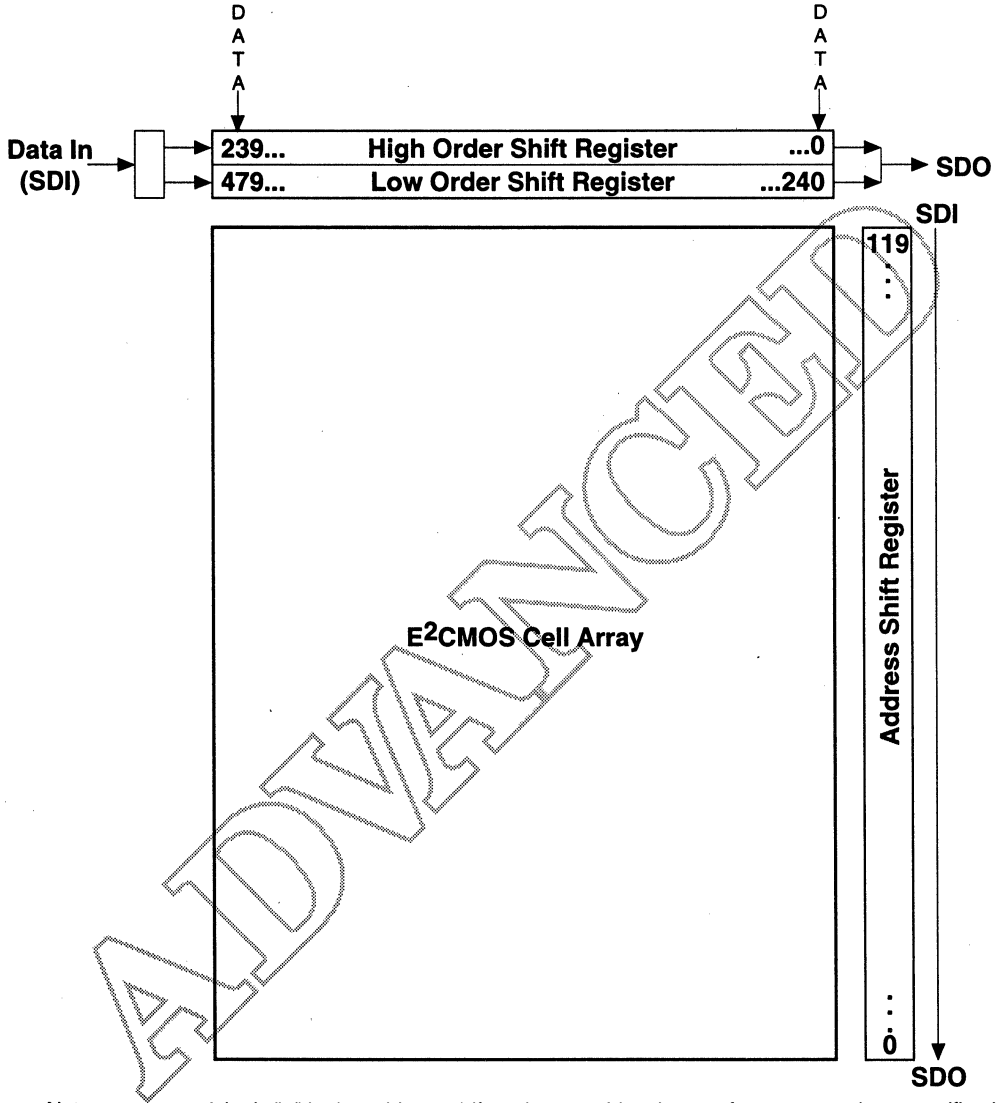


Figure 18. ispLSI Device & Shift Register Layout



Note: A logic "1" in the address shift register enables the row for programming or verification. A logic "0" disables it.

Programming Voltage/Timing Specifications

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.	MAX.	UNITS
V_{CCP}	Programmir		4.75	5	5.25	V
I_{CCP}	Programming Supply Current ispEN		–	50	100	mA
V_{IHP}	Input Voltage High		2.0	–	V_{CCP}	V
V_{ILP}	Input Vopltage Low		0	–	0.8	V
I_{IP}	Input Current		–	100	200	µA
V_{OHP}	Output Volatge High	I_{OH} = -3.2 mA	2.4	–	V_{CCP}	V
V_{OLP}	Output Voltage Low	I_{OL} =5 mA	0	–	0.5	V
t_d	Pulse Sequence Delay		1	5	10	µs
t_{isp}	$\overline{ispEN}$ to Output 3-State		–	1	10	µs
t_{su}	Setup Time		.1	.5	–	µs
t_h	Hold Time		.1	.5	–	µs
t_{clk}	Clock Pulse Width		0.5	1	–	µs
t_{pwv}	Verify Pulse Width		20	30	–	µs
t_{pwp}	Programming Pulse Width		40	–	100	ms
t_{bew}	Bulk Erase Pulse Width		200	–	–	ms
t_{rst}	Reset Time From Valid V_{CCP}		45	–	–	µs

Figure 19. Timing Waveform for isp Operation

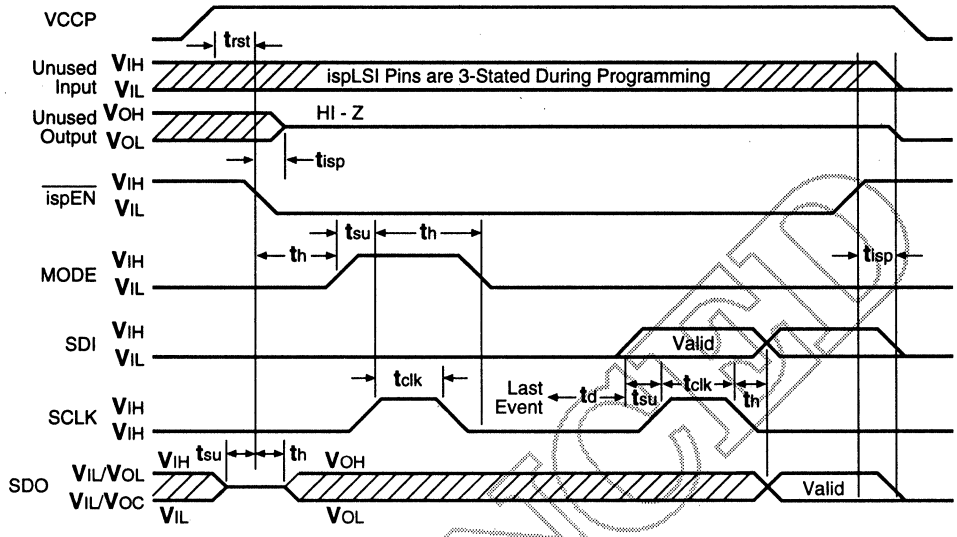
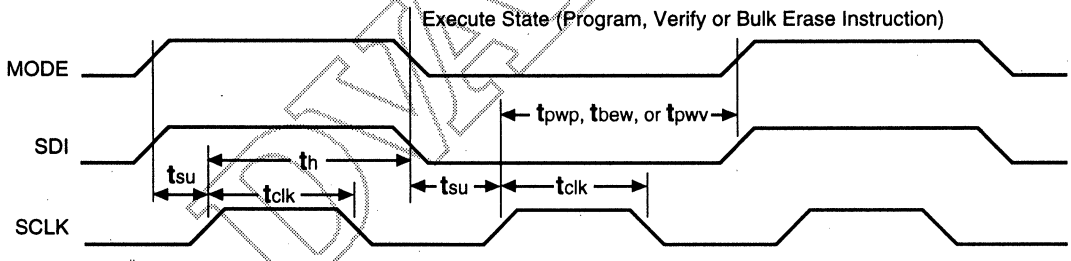
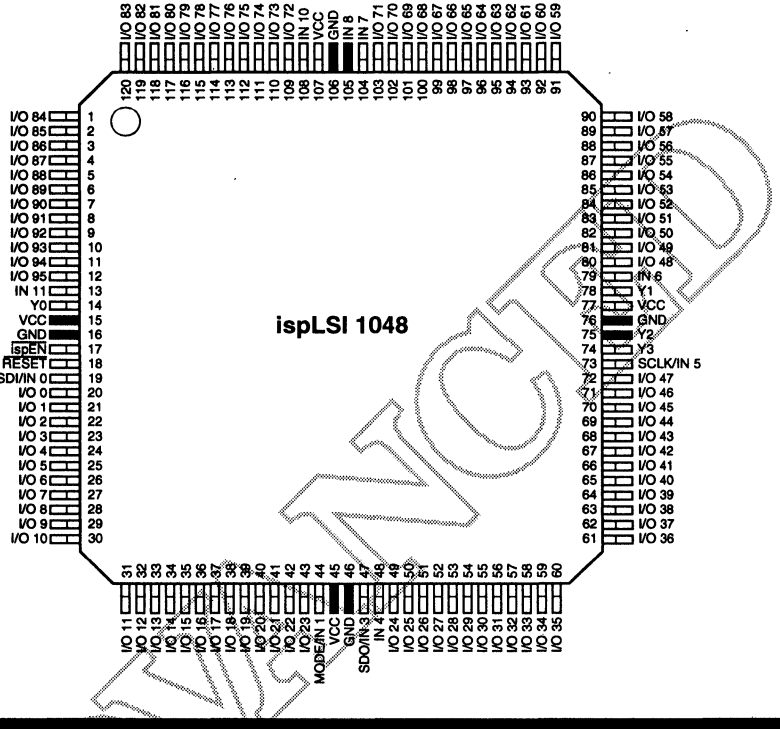


Figure 20. Program, Verify & Bulk Erase Waveform

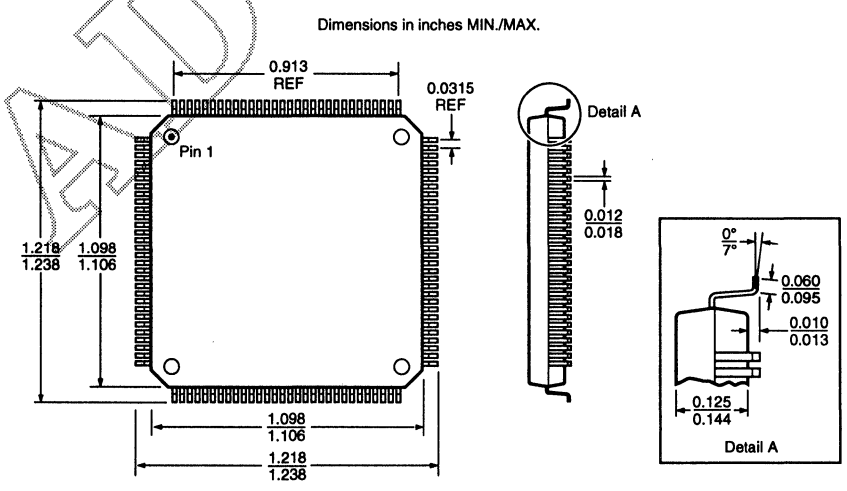


Pin Configuration

ispLSI 1048 PQFP Pinout Diagram



120-Pin PQFP



Section 1: Introduction to pLSI™ and ispLSI™

Section 2: pLSI and ispLSI Data Sheets

Section 3: GAL® Data Specifications

GAL Product Index 3-ii

Introduction to Generic Array Logic™ 3-1

GAL16V8A/B 3-3

GAL20V8A/B 3-9

GAL18V10 3-15

GAL22V10/B 3-19

GAL26CV12 3-25

GAL20RA10 3-29

GAL20XV10B 3-33

GAL6001 3-37

ispGAL16Z8 3-41

Military GAL Device Summary 3-45

Section 4: pLSI and ispLSI Architecture

Section 5: pLSI and ispLSI Advantages

Section 6: pLSI and ispLSI Software Development Tools

Section 7: pLSI and ispLSI Design Optimization

Section 8: pLSI and ispLSI Application Notes

Section 9: pLSI and ispLSI Article Reprints

Section 10: pLSI and ispLSI Quality, Reliability & Programmer Support

Section 11: Sales Information

GAL Product Index

Commercial Grade Devices

DEVICE	PINS	t_{PD} (ns)	I_{CC} (mA)	DESCRIPTION	PAGE
GAL16V8A/B	20	7.5, 10, 15, 25	55, 90, 115	E²CMOS Generic PLD	3-3
GAL20V8A/B	24	7.5, 10, 15, 25	55, 90, 115	E²CMOS Generic PLD	3-9
GAL18V10	20	15, 20	115	E²CMOS Universal PLD	3-15
GAL22V10/B	24	10, 15, 25	130	E²CMOS Universal PLD	3-19
GAL26CV12	28	15, 20	130	E²CMOS Universal PLD	3-25
GAL20RA10	24	12, 15, 20, 30	100	E²CMOS Asynchronous PLD	3-29
GAL20XV10B	24	10, 15	90	E²CMOS Exclusive-OR PLD	3-33
GAL6001	24	30, 35	150	E²CMOS FPLA	3-37
ispGAL16Z8	24	20, 25	90	E²CMOS In-System-Programmable PLD	3-41

Industrial Grade Devices

DEVICE	PINS	t_{PD} (ns)	I_{CC} (mA)	DESCRIPTION	PAGE
GAL16V8A/B	20	10, 15, 20, 25	65, 130	E²CMOS Generic PLD	See GAL Data Book
GAL20V8A/B	24	10, 15, 20, 25	65, 130	E²CMOS Generic PLD	See GAL Data Book
GAL18V10	20	20	125	E²CMOS Universal PLD	See GAL Data Book
GAL22V10/B	24	15, 20, 25	150	E²CMOS Universal PLD	See GAL Data Book
GAL26CV12	28	20	150	E²CMOS Universal PLD	See GAL Data Book
GAL20RA10	24	20	120	E²CMOS Asynchronous PLD	See GAL Data Book

MIL-STD-883C Grade Devices

DEVICE	PINS	t_{PD} (ns)	I_{CC} (mA)	DESCRIPTION	PAGE
GAL16V8A/B	20	10, 15, 20, 25, 30	65, 130	E²CMOS Universal PLD	See GAL Data Book
GAL20V8A	24	15, 20, 25, 30	65, 130	E²CMOS Universal PLD	See GAL Data Book
GAL22V10	24	15, 20, 25, 30	150	E²CMOS Universal PLD	See GAL Data Book
GAL26CV12	28	20, 25	160	E²CMOS Universal PLD	See GAL Data Book
GAL20RA10	24	20, 25	120	E²CMOS Asynchronous PLD	See GAL Data Book

Introduction to Generic Array Logic

Introduction

Lattice Semiconductor, located in Hillsboro, Oregon, was founded in 1983 to design, develop and manufacture high-performance semiconductor components. It is a firm belief at Lattice that technological evolution can be accelerated through the continued development of higher-speed and architecturally superior products.

GAL devices are ideal for four important reasons:

1. GAL devices have inherently superior quality and reliability.
2. GAL devices can directly replace PAL devices in nearly every application.
3. GAL devices have the low power consumption of CMOS, one-fourth to one-half that of bipolar devices.
4. GAL devices utilize Output Logic Macrocells (OLMCs), which allow the user to configure outputs as needed.

The GAL Concept

E²CMOS — The Ideal Technology

Of the three major technologies available for producing PLDs, the technology of choice is clearly E²CMOS. E²CMOS offers testability, quality, high speed, low power, and instant erasure.

Testability

The biggest advantage of E²CMOS over competing technologies is its inherent testability. Capitalizing on very fast (100ms) erase times, Lattice repeatedly patterns and erases all devices during manufacture. Lattice tests each GAL device for AC, DC, and functional characteristics. The result is guaranteed 100% programming and functional yields.

Low Power

Another advantage of E²CMOS technology is the low power consumption of CMOS. CMOS provides users the immediate benefit of decreased system power requirements allowing for higher reliability and cooler running systems. Low power CMOS technology also permits circuit designs of much higher functional density, because of lower junction temperatures and power requirements on chip. The user benefits because higher functional density means further reduction of chip count and smaller boards in the system.

High Speed

Also advantageous is the very high speed attainable with Lattice's state-of-the-art E²CMOS process. Lattice GAL devices are as fast or faster than bipolar and UVCMOS PLDs.

Prototyping and Error Recovery

Finally, E²CMOS gives the user instant erasability with no additional handling or special packages necessary. This provides ideal products for prototyping because designs can be revised instantly, with no waste and no waiting. On the manufacturing floor instant erasability can also be a big advantage for dealing with pattern changes or error recovery. If a GAL device is accidentally programmed to the wrong pattern, simply reprogram the device. No other technology offers this advantage.

A Look at Other Technologies

Here, the technologies that compete with E²CMOS — bipolar and UVCMOS — are compared with the E²CMOS approach.

Bipolar

Bipolar fuse-link technology was the first available for programmable logic devices. Although it offers high speed, it is saddled with high power dissipation. High power dissipation increases your system power supply and cooling requirements, and limits the functional density of bipolar devices.

Another weakness of this technology is the one-time-programmable fuses. Complete testing of bipolar PLDs is impossible because the fuse array cannot be tested before programming. Bipolar PLD manufacturers must rely on complex schemes using test rows and columns to simulate and correlate their device's performance. The result is programming failures at the customer location. Any misprogrammed devices due to mistakes during prototyping or errors on the production floor must be discarded because bipolar PLDs cannot be reprogrammed.

UVCMOS

UVCMOS addresses many weaknesses of the bipolar approach but introduces many shortcomings of its own. This technology requires less power and is reprogrammable, but reprogrammability comes at the expense of slower speeds.

Testability is increased over bipolar since the "fuse" array can be programmed and tested by the manufacturer. The problem here is the long (20 minutes) erase times coupled with the requirement of exposing the devices to ultraviolet light for erasing. This becomes a very expensive step in the manufacturing process. Because of the time involved, patterning and erasing is performed only once — a compromised rather than complete functional test.

Additionally, the devices must be housed in expensive windowed packages to allow users to erase them. Again, programming these devices is time-consuming and cumbersome due to the 20-minute UV exposure required to erase them. As a cost-cutting measure, UVCMOS PLD manufacturers offer their devices in windowless packages. Although windowless packages are less expensive, they cannot be completely tested or reprogrammed. These factors significantly detract from the desirability of this technology.

The GAL Advantage

GAL devices are ideal programmable logic devices because, as the name implies, they are architecturally generic. Lattice has employed the macrocell approach, which allows users to define the architecture and functionality of each output. The key benefit to the user is the freedom from being restricted to any specific architecture. This is advantageous at both the manufacturing level and the design level.

Design Advantages

Early programmable logic devices gave the user the ability to specify a function, but limited them to specific, predetermined output architectures. Comparing the GAL device with fixed-architecture programmable logic devices is much like comparing these same fixed PLDs with SSI/MSI devices. The GAL family is the next generation in simplified system design. The user does not have to search for the architecture that best suits a particular design. Instead, the GAL family's generic architecture lets him configure as he goes.

Manufacturing Advantages

The one-device-does-all approach greatly simplifies manufacturing flow. Inventorying one generic-architecture GAL device type versus having to monitor and maintain many different device types, saves money and minimizes paperwork. Manufacturing flow is much smoother because the handling process is greatly simplified. A generic architecture GAL device also reduces the risk of running out of inventory and halting production, which can be very expensive. Reduced chance of obsolete inventory and easier QA tracking are additional benefits of the generic architecture.

The Ideal Package

Programmable logic devices are ideal for designing today's systems. Lattice Semiconductor believes that the ideal design approach should be supported with the ideal products. It was on this premise that GAL devices were invented. The ideal device—with a generic architecture—fabricated with the ideal process technology, E²CMOS.

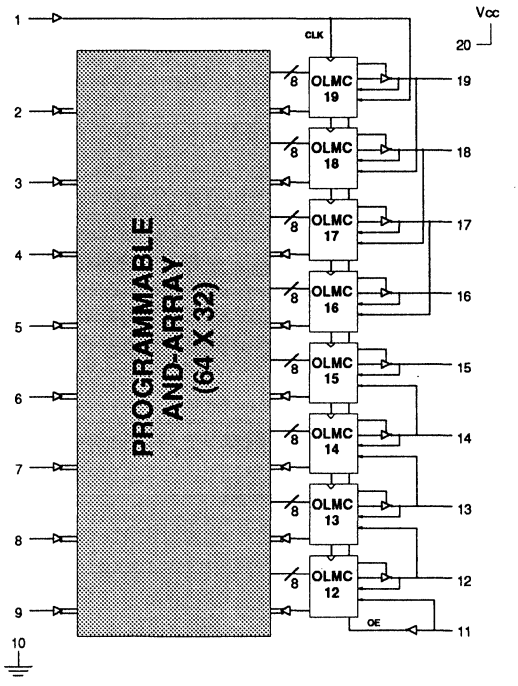
GAL16V8B
GAL16V8A

High Performance E²CMOS PLD

FEATURES

- **HIGH PERFORMANCE E²CMOS® TECHNOLOGY**
  - **— 7.5 ns Maximum Propagation Delay**
  - **— Fmax = 100 MHz**
  - **— 5 ns Maximum from Clock Input to Data Output**
  - **— TTL Compatible 24 mA Outputs**
  - **— UltraMOS® Advanced CMOS Technology**
- **50% to 75% REDUCTION IN POWER FROM BIPOLAR**
  - **— 75mA Typ Icc on Low Power Device**
  - **— 45mA Typ Icc on Quarter Power Device**
- **ACTIVE PULL-UPS ON ALL PINS (GAL16V8B)**
- **E² CELL TECHNOLOGY**
  - **— Reconfigurable Logic**
  - **— Reprogrammable Cells**
  - **— 100% Tested/Guaranteed 100% Yields**
  - **— High Speed Electrical Erasure (<100ms)**
  - **— 20 Year Data Retention**
- **EIGHT OUTPUT LOGIC MACROCELLS**
  - **— Maximum Flexibility for Complex Logic Designs**
  - **— Programmable Output Polarity**
  - **— Also Emulates 20-pin PAL® Devices with Full Function/Fuse Map/Parametric Compatibility**
- **PRELOAD AND POWER-ON RESET OF ALL REGISTERS**
  - **— 100% Functional Testability**
- **APPLICATIONS INCLUDE:**
  - **— DMA Control**
  - **— State Machine Control**
  - **— High Speed Graphics Processing**
  - **— Standard Logic Speed Upgrade**
- **ELECTRONIC SIGNATURE FOR IDENTIFICATION**

FUNCTIONAL BLOCK DIAGRAM



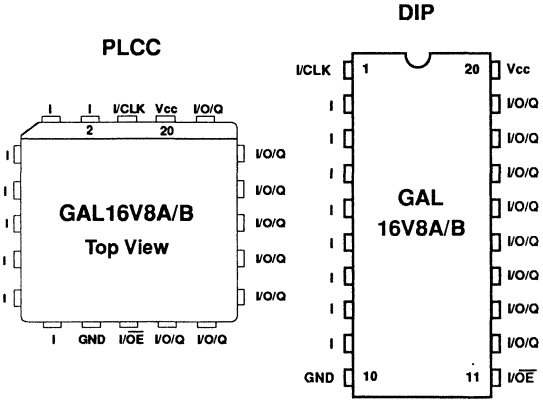
DESCRIPTION

The GAL16V8B, at 7.5 ns maximum propagation delay time, combines a high performance CMOS process with Electrically Erasable (E^2) floating gate technology to provide the highest speed performance available in the PLD market. High speed erase times (<100ms) allow the devices to be reprogrammed quickly and efficiently.

The generic architecture provides maximum design flexibility by allowing the Output Logic Macrocell (OLMC) to be configured by the user. An important subset of the many architecture configurations possible with the GAL16V8A/B are the PAL architectures listed in the table of the macrocell description section. GAL16V8A/B devices are capable of emulating any of these PAL architectures with full function/fuse map/parametric compatibility.

Unique test circuitry and reprogrammable cells allow complete AC, DC, and functional testing during manufacture. As a result, LATTICE is able to guarantee 100% field programmability and functionality of all GAL® products. LATTICE also guarantees 100 erase/rewrite cycles and data retention in excess of 20 years.

PIN CONFIGURATION



LATTICE SEMICONDUCTOR CORP., 5555 N.E. Moore Ct., Hillsboro, Oregon 97124, U.S.A.
Tel. (503) 681-0118; 1-800-FASTGAL; FAX (503) 681-3037

December 1991

ABSOLUTE MAXIMUM RATINGS[1]

Supply voltage V_{CC} .....–0.5 to +7V
Input voltage applied .....–2.5 to V_{CC} +1.0V
Off-state output voltage applied .....–2.5 to V_{CC} +1.0V
Storage Temperature .....–65 to 150°C
Ambient Temperature with
Power Applied .....–55 to 125°C

1. Stresses above those listed under the "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress only ratings and functional operation of the device at these or at any other conditions above those indicated in the operational sections of this specification is not implied (while programming, follow the programming specifications).

RECOMMENDED OPERATING COND.

Commercial Devices:
Ambient Temperature (T_A) .....0 to 75°C
Supply voltage (V_{CC})
with Respect to Ground .....+4.75 to +5.25V

DC ELECTRICAL CHARACTERISTICS

Over Recommended Operating Conditions (Unless Otherwise Specified)

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.[3]	MAX.	UNITS
V_{IL}	Input Low Voltage		Vss – 0.5	—	0.8	V
V_{IH}	Input High Voltage		2.0	—	Vcc+1	V
I_{IL}[1]	Input or I/O Low Leakage Current	$0V \le V_{IN} \le V_{IL}$ (MAX.)	—	—	–100	µA
I_{IH}	Input or I/O High Leakage Current	$3.5V \le V_{IN} \le V_{CC}$	—	—	10	µA
V_{OL}	Output Low Voltage	I_{OL} = MAX. Vin = V_{IL} or V_{IH}	—	—	0.5	V
V_{OH}	Output High Voltage	I_{OH} = MAX. Vin = V_{IL} or V_{IH}	2.4	—	—	V
I_{OL}	Low Level Output Current		—	—	24	mA
I_{OH}	High Level Output Current		—	—	–3.2	mA
I_{OS}[2]	Output Short Circuit Current	V_{CC} = 5V V_{OUT} = 0.5V T_A = 25°C	–30	—	–150	mA
I_{CC}	Operating Power Supply Current	V_{IL} = 0.5V V_{IH} = 3.0V f_{toggle} = 25MHz Outputs Open (no load)	—	75	115	mA

1) The leakage current is due to the internal pull-up resistor on all pins. See **Input Buffer** section for more information.
2) One output at a time for a maximum duration of one second. Vout = 0.5V was selected to avoid test problems caused by tester ground degradation. Guaranteed but not 100% tested.
3) Typical values are at Vcc = 5V and T_A = 25 °C

CAPACITANCE (T_A = 25°C, f = 1.0 MHz)

SYMBOL	PARAMETER	MAXIMUM*	UNITS	TEST CONDITIONS
C_I	Input Capacitance	8	pF	V_{CC} = 5.0V, V_I = 2.0V
$C_{I/O}$	I/O Capacitance	8	pF	V_{CC} = 5.0V, $V_{I/O}$ = 2.0V

*Guaranteed but not 100% tested.

AC SWITCHING CHARACTERISTICS

Over Recommended Operating Conditions

PARAMETER	TEST COND[1].	DESCRIPTION		-7		-10		UNITS
				MIN.	MAX.	MIN.	MAX.	
tpd	1	Input or I/O to Combinational Output	8 outputs switching	3	7.5	3	10	ns
			1 output switching	—	7	—	—	ns
tco	1	Clock to Output Delay		2	5	2	7	ns
tcf[2]	—	Clock to Feedback Delay		—	3	—	6	ns
tsu	—	Setup Time, Input or Feedback before Clock↑		7	—	10	—	ns
th	—	Hold Time, Input or Feedback after Clock↑		0	—	0	—	ns
fmax[3]	1	Maximum Clock Frequency with External Feedback, 1/(tsu + tco)		83.3	—	58.8	—	MHz
	1	Maximum Clock Frequency with Internal Feedback, 1/(tsu + tcf)		100	—	62.5	—	MHz
	1	Maximum Clock Frequency with No Feedback		100	—	62.5	—	MHz
twh[4]	—	Clock Pulse Duration, High		5	—	8	—	ns
twl[4]	—	Clock Pulse Duration, Low		5	—	8	—	ns
ten	2	Input or I/O to Output		3	9	3	10	ns
	2	OE↓ to Output		2	6	2	10	ns
tdis	3	Input or I/O to Output		2	9	2	10	ns
	3	OE↑ to Output		1.5	6	1.5	10	ns

1) Refer to **Switching Test Conditions** section.
2) Calculated from fmax with internal feedback. Refer to **fmax Descriptions** section.
3) Refer to **fmax Descriptions** section.
4) Clock pulses of widths less than the specification may be detected as valid clock signals.

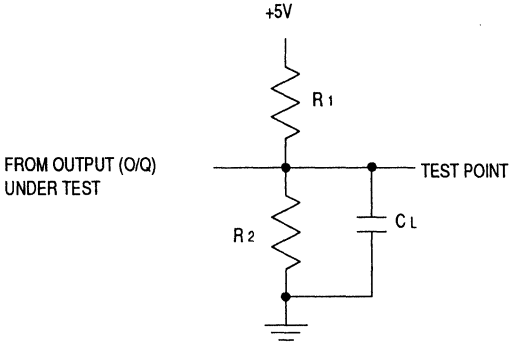
SWITCHING TEST CONDITIONS

Input Pulse Levels	GND to 3.0V
Input Rise and Fall Times	3ns 10% – 90%
Input Timing Reference Levels	1.5V
Output Timing Reference Levels	1.5V
Output Load	See Figure

3-state levels are measured 0.5V from steady-state active level.

Output Load Conditions (see figure)

Test Condition		R1	R2	CL
1		200Ω	390Ω	50pF
2	Active High	∞	390Ω	50pF
	Active Low	200Ω	390Ω	50pF
3	Active High	∞	390Ω	5pF
	Active Low	200Ω	390Ω	5pF



C L INCLUDES JIG AND PROBE TOTAL CAPACITANCE

ABSOLUTE MAXIMUM RATINGS[1]

Supply voltage V_{CC} .....–0.5 to +7V
Input voltage applied .....–2.5 to V_{CC} +1.0V
Off-state output voltage applied .....–2.5 to V_{CC} +1.0V
Storage Temperature .....–65 to 150°C
Ambient Temperature with
Power Applied .....–55 to 125°C

1. Stresses above those listed under the "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress only ratings and functional operation of the device at these or at any other conditions above those indicated in the operational sections of this specification is not implied (while programming, follow the programming specifications).

RECOMMENDED OPERATING COND.

Commercial Devices:
Ambient Temperature (T_A) .....0 to 75°C
Supply voltage (V_{CC})
with Respect to Ground .....+4.75 to +5.25V

DC ELECTRICAL CHARACTERISTICS

Over Recommended Operating Conditions (Unless Otherwise Specified)

SYMBOL	PARAMETER	CONDITION			MIN.	TYP.[2]	MAX.	UNITS
V_{IL}	Input Low Voltage				Vss – 0.5	—	0.8	V
V_{IH}	Input High Voltage				2.0	—	Vcc+1	V
I_{IL}	Input or I/O Low Leakage Current	$0V \le V_{IN} \le V_{IL}$ (MAX.)			—	—	-10	µA
I_{IH}	Input or I/O High Leakage Current	$V_{IH} \le V_{IN} \le V_{CC}$			—	—	10	µA
V_{OL}	Output Low Voltage	I_{OL} = MAX. Vin = V_{IL} or V_{IH}			—	—	0.5	V
V_{OH}	Output High Voltage	I_{OH} = MAX. Vin = V_{IL} or V_{IH}			2.4	—	—	V
I_{OL}	Low Level Output Current				—	—	24	mA
I_{OH}	High Level Output Current				—	—	–3.2	mA
I_{OS}[1]	Output Short Circuit Current	V_{CC} = 5V	V_{OUT} = 0.5V	T_A = 25°C	–30	—	–150	mA
I_{CC}	Operating Power Supply Current	V_{IL} = 0.5V V_{IH} = 3.0V Outputs Open (no load)	f_{toggle} = 15MHz	**L** -25	—	75	90	mA
			f_{toggle} = 25MHz	**L** -10/-15	—	75	115	mA
			f_{toggle} = 15MHz	**Q** -15/-25	—	45	55	mA

1) One output at a time for a maximum duration of one second. Vout = 0.5V was selected to avoid test problems caused by tester ground degradation. Guaranteed but not 100% tested.
2) Typical values are at Vcc = 5V and T_A = 25 °C

CAPACITANCE (T_A = 25°C, f = 1.0 MHz)

SYMBOL	PARAMETER	MAXIMUM*	UNITS	TEST CONDITIONS
C_I	Input Capacitance	8	pF	V_{CC} = 5.0V, V_I = 2.0V
$C_{I/O}$	I/O Capacitance	10	pF	V_{CC} = 5.0V, $V_{I/O}$ = 2.0V

*Guaranteed but not 100% tested.

AC SWITCHING CHARACTERISTICS

Over Recommended Operating Conditions

PARAMETER	TEST COND[1].	DESCRIPTION	-10		-15		-25		UNITS
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
tpd	1	Input or I/O to Combinational Output	3	10	3	15	3	25	ns
tco	1	Clock to Output Delay	2	7	2	10	2	12	ns
tcf[2]	—	Clock to Feedback Delay	—	7	—	8	—	10	ns
tsu	—	Setup Time, Input or Feedback before Clock↑	10	—	12	—	15	—	ns
th	—	Hold Time, Input or Feedback after Clock↑	0	—	0	—	0	—	ns
fmax[3]	1	Maximum Clock Frequency with External Feedback, 1/(tsu + tco)	58.8	—	45.5	—	37	—	MHz
	1	Maximum Clock Frequency with Internal Feedback, 1/(tsu + tcf)	58.8	—	50	—	40	—	MHz
	1	Maximum Clock Frequency with No Feedback	62.5	—	62.5	—	41.6	—	MHz
twh[4]	—	Clock Pulse Duration, High	8	—	8	—	12	—	ns
twl[4]	—	Clock Pulse Duration, Low	8	—	8	—	12	—	ns
ten	2	Input or I/O to Output Enabled	—	10	—	15	—	25	ns
	2	OE↓ to Output Enabled	—	10	—	15	—	20	ns
tdis	3	Input or I/O to Output Disabled	—	10	—	15	—	25	ns
	3	OE↑ to Output Disabled	—	10	—	15	—	20	ns

1) Refer to **Switching Test Conditions** section.
2) Calculated from **fmax** with internal feedback. Refer to **fmax Descriptions** section.
3) Refer to **fmax Descriptions** section.
4) Clock pulses of widths less than the specification may be detected as valid clock signals.

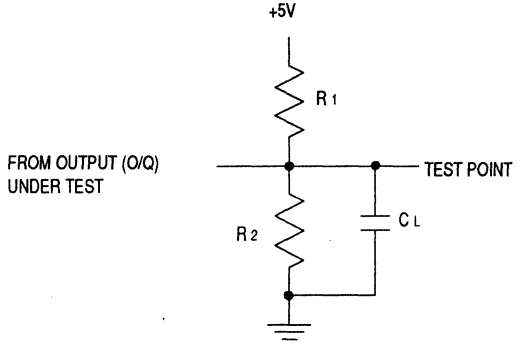
SWITCHING TEST CONDITIONS

Input Pulse Levels	GND to 3.0V
Input Rise and Fall Times	3ns 10% – 90%
Input Timing Reference Levels	1.5V
Output Timing Reference Levels	1.5V
Output Load	See Figure

3-state levels are measured 0.5V from steady-state active level.

Output Load Conditions (see figure)

Test Condition		R1	R2	CL
1		200Ω	390Ω	50pF
2	Active High	∞	390Ω	50pF
	Active Low	200Ω	390Ω	50pF
3	Active High	∞	390Ω	5pF
	Active Low	200Ω	390Ω	5pF



C L INCLUDES JIG AND PROBE TOTAL CAPACITANCE

GAL16V8A/B ORDERING INFORMATION

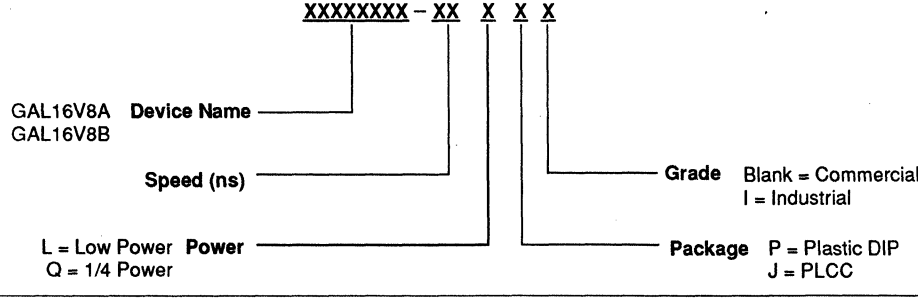
Commercial Grade Specifications

Tpd (ns)	Tsu (ns)	Tco (ns)	Icc (mA)	Ordering #	Package
7.5	7	5	115	GAL16V8B-7LP	20-Pin Plastic DIP
			115	GAL16V8B-7LJ	20-Lead PLCC
10	10	7	115	GAL16V8B-10LP	20-Pin Plastic DIP
			115	GAL16V8B-10LJ	20-Lead PLCC
			115	GAL16V8A-10LP	20-Pin Plastic DIP
			115	GAL16V8A-10LJ	20-Lead PLCC
15	12	10	55	GAL16V8A-15QP	20-Pin Plastic DIP
			55	GAL16V8A-15QJ	20-Lead PLCC
			115	GAL16V8A-15LP	20-Pin Plastic DIP
			115	GAL16V8A-15LJ	20-Lead PLCC
25	15	12	55	GAL16V8A-25QP	20-Pin Plastic DIP
			55	GAL16V8A-25QJ	20-Lead PLCC
			90	GAL16V8A-25LP	20-Pin Plastic DIP
			90	GAL16V8A-25LJ	20-Lead PLCC

Industrial Grade Specifications

Tpd (ns)	Tsu (ns)	Tco (ns)	Icc (mA)	Ordering #	Package
10	10	7	130	GAL16V8B-10LPI	20-Pin Plastic DIP
			130	GAL16V8B-10LJI	20-Lead PLCC
15	12	10	130	GAL16V8B-15LPI	20-Pin Plastic DIP
			130	GAL16V8B-15LJI	20-Lead PLCC
			130	GAL16V8A-15LPI	20-Pin Plastic DIP
			130	GAL16V8A-15LJI	20-Lead PLCC
20	13	11	65	GAL16V8A-20QPI	20-Pin Plastic DIP
			65	GAL16V8A-20QJI	20-Lead PLCC
25	15	12	65	GAL16V8A-25QPI	20-Pin Plastic DIP
			65	GAL16V8A-25QJI	20-Lead PLCC
			130	GAL16V8A-25LPI	20-Pin Plastic DIP
			130	GAL16V8A-25LJI	20-Lead PLCC

PART NUMBER DESCRIPTION



GAL20V8B
GAL20V8A

High Performance E²CMOS PLD

FEATURES

- **HIGH PERFORMANCE E²CMOS® TECHNOLOGY**
  - **— 7.5 ns Maximum Propagation Delay**
  - **— Fmax = 100 MHz**
  - **— 5 ns Maximum from Clock Input to Data Output**
  - **— TTL Compatible 24 mA Outputs**
  - **— UltraMOS® Advanced CMOS Technology**
- **50% to 75% REDUCTION IN POWER FROM BIPOLAR**
  - **— 75mA Typ Icc on Low Power Device**
  - **— 45mA Typ Icc on Quarter Power Device**
- **ACTIVE PULL-UPS ON ALL PINS (GAL20V8B)**
- **E² CELL TECHNOLOGY**
  - **— Reconfigurable Logic**
  - **— Reprogrammable Cells**
  - **— 100% Tested/Guaranteed 100% Yields**
  - **— High Speed Electrical Erasure (<100ms)**
  - **— 20 Year Data Retention**
- **EIGHT OUTPUT LOGIC MACROCELLS**
  - **— Maximum Flexibility for Complex Logic Designs**
  - **— Programmable Output Polarity**
  - **— Also Emulates 24-pin PAL® Devices with Full Function/Fuse Map/Parametric Compatibility**
- **PRELOAD AND POWER-ON RESET OF ALL REGISTERS**
  - **— 100% Functional Testability**
- **APPLICATIONS INCLUDE:**
  - **— DMA Control**
  - **— State Machine Control**
  - **— High Speed Graphics Processing**
  - **— Standard Logic Speed Upgrade**
- **ELECTRONIC SIGNATURE FOR IDENTIFICATION**

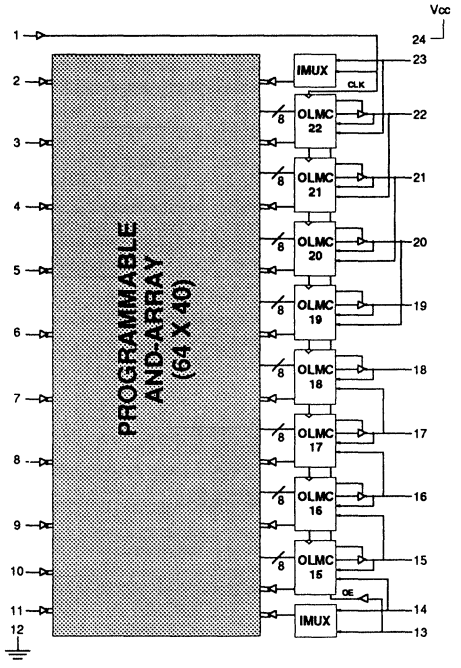
DESCRIPTION

The GAL20V8B, at 7.5ns maximum propagation delay time, combines a high performance CMOS process with Electrically Erasable (E^2) floating gate technology to provide the highest speed performance available in the PLD market. High speed erase times (<100ms) allow the devices to be reprogrammed quickly and efficiently.

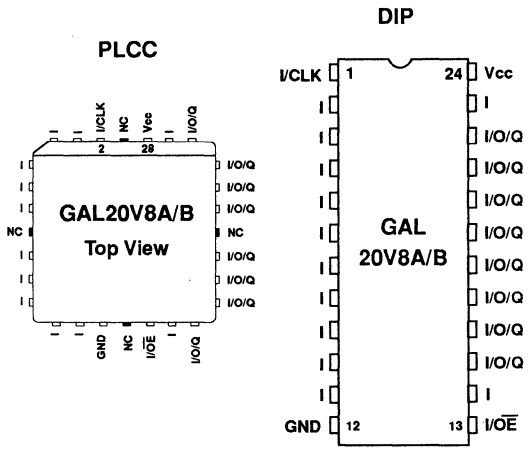
The generic architecture provides maximum design flexibility by allowing the Output Logic Macrocell (OLMC) to be configured by the user. An important subset of the many architecture configurations possible with the GAL20V8A/B are the PAL architectures listed in the table of the macrocell description section. GAL20V8A/B devices are capable of emulating any of these PAL architectures with full function/fuse map/parametric compatibility.

Unique test circuitry and reprogrammable cells allow complete AC, DC, and functional testing during manufacture. As a result, LATTICE is able to guarantee 100% field programmability and functionality of all GAL® products. LATTICE also guarantees 100 erase/rewrite cycles and data retention in excess of 20 years.

FUNCTIONAL BLOCK DIAGRAM



PIN CONFIGURATION



LATTICE SEMICONDUCTOR CORP., 5555 N.E. Moore Ct., Hillsboro, Oregon 97124, U.S.A.
Tel. (503) 681-0118; 1-800-FASTGAL; FAX (503) 681-3037

December 1991

ABSOLUTE MAXIMUM RATINGS[1]

Supply voltage V_{CC} .....–0.5 to +7V
Input voltage applied .....–2.5 to V_{CC} +1.0V
Off-state output voltage applied .....–2.5 to V_{CC} +1.0V
Storage Temperature .....–65 to 150°C
Ambient Temperature with
Power Applied .....–55 to 125°C

1. Stresses above those listed under the "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress only ratings and functional operation of the device at these or at any other conditions above those indicated in the operational sections of this specification is not implied (while programming, follow the programming specifications).

RECOMMENDED OPERATING COND.

Commercial Devices:
Ambient Temperature (T_A) .....0 to 75°C
Supply voltage (V_{CC})
with Respect to Ground .....+4.75 to +5.25V

DC ELECTRICAL CHARACTERISTICS

Over Recommended Operating Conditions (Unless Otherwise Specified)

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.[3]	MAX.	UNITS
V_{IL}	Input Low Voltage		Vss – 0.5	—	0.8	V
V_{IH}	Input High Voltage		2.0	—	Vcc+1	V
I_{IL}[1]	Input or I/O Low Leakage Current	$0V \le V_{IN} \le V_{IL}$ (MAX.)	—	—	–100	µA
I_{IH}	Input or I/O High Leakage Current	$3.5V \le V_{IN} \le V_{CC}$	—	—	10	µA
V_{OL}	Output Low Voltage	I_{OL} = MAX. Vin = V_{IL} or V_{IH}	—	—	0.5	V
V_{OH}	Output High Voltage	I_{OH} = MAX. Vin = V_{IL} or V_{IH}	2.4	—	—	V
I_{OL}	Low Level Output Current		—	—	24	mA
I_{OH}	High Level Output Current		—	—	–3.2	mA
I_{OS}[2]	Output Short Circuit Current	V_{CC} = 5V V_{OUT} = 0.5V T_A = 25°C	–30	—	–150	mA
I_{CC}	Operating Power Supply Current	V_{IL} = 0.5V V_{IH} = 3.0V f_{toggle} = 25MHz Outputs Open (no load)	—	75	115	mA

1) The leakage current is due to the internal pull-up resistor on all pins. See **Input Buffer** section for more information.
2) One output at a time for a maximum duration of one second. Vout = 0.5V was selected to avoid test problems caused by tester ground degradation. Guaranteed but not 100% tested.
3) Typical values are at Vcc = 5V and T_A = 25 °C

CAPACITANCE (T_A = 25°C, f = 1.0 MHz)

SYMBOL	PARAMETER	MAXIMUM*	UNITS	TEST CONDITIONS
C_I	Input Capacitance	8	pF	V_{CC} = 5.0V, V_I = 2.0V
$C_{I/O}$	I/O Capacitance	8	pF	V_{CC} = 5.0V, $V_{I/O}$ = 2.0V

*Guaranteed but not 100% tested.

AC SWITCHING CHARACTERISTICS

Over Recommended Operating Conditions

PARAMETER	TEST COND[1].	DESCRIPTION		-7		-10		UNITS
				MIN.	MAX.	MIN.	MAX.	
tpd	1	Input or I/O to Combinational Output	8 outputs switching	3	7.5	3	10	ns
			1 output switching	—	7	—	—	ns
tco	1	Clock to Output Delay		2	5	2	7	ns
tcf[2]	—	Clock to Feedback Delay		—	3	—	6	ns
tsu	—	Setup Time, Input or Feedback before Clock↑		7	—	10	—	ns
th	—	Hold Time, Input or Feedback after Clock↑		0	—	0	—	ns
fmax[3]	1	Maximum Clock Frequency with External Feedback, 1/(tsu + tco)		83.3	—	58.8	—	MHz
	1	Maximum Clock Frequency with Internal Feedback, 1/(tsu + tcf)		100	—	62.5	—	MHz
	1	Maximum Clock Frequency with No Feedback		100	—	62.5	—	MHz
twh[4]	—	Clock Pulse Duration, High		5	—	8	—	ns
twl[4]	—	Clock Pulse Duration, Low		5	—	8	—	ns
ten	2	Input or I/O to Output		3	9	3	10	ns
	2	OE↓ to Output		2	6	2	10	ns
tdis	3	Input or I/O to Output		2	9	2	10	ns
	3	OE↑ to Output		1.5	6	1.5	10	ns

1) Refer to **Switching Test Conditions** section.
2) Calculated from fmax with internal feedback. Refer to **fmax Descriptions** section.
3) Refer to **fmax Descriptions** section.
4) Clock pulses of widths less than the specification may be detected as valid clock signals.

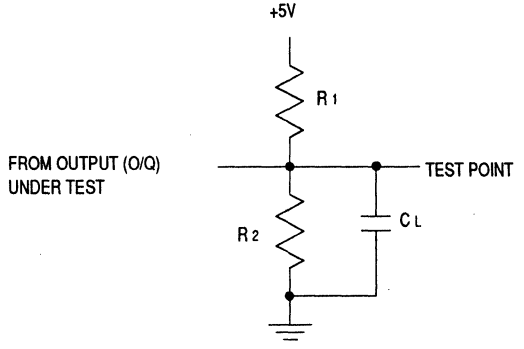
SWITCHING TEST CONDITIONS

Input Pulse Levels	GND to 3.0V
Input Rise and Fall Times	3ns 10% – 90%
Input Timing Reference Levels	1.5V
Output Timing Reference Levels	1.5V
Output Load	See Figure

3-state levels are measured 0.5V from steady-state active level.

Output Load Conditions (see figure)

Test Condition		R1	R2	CL
1		200Ω	390Ω	50pF
2	Active High	∞	390Ω	50pF
	Active Low	200Ω	390Ω	50pF
3	Active High	∞	390Ω	5pF
	Active Low	200Ω	390Ω	5pF



C L INCLUDES JIG AND PROBE TOTAL CAPACITANCE

ABSOLUTE MAXIMUM RATINGS(1)

Supply voltage V_{CC} .....–0.5 to +7V
Input voltage applied .....–2.5 to V_{CC} +1.0V
Off-state output voltage applied .....–2.5 to V_{CC} +1.0V
Storage Temperature .....–65 to 150°C
Ambient Temperature with
Power Applied .....–55 to 125°C

1. Stresses above those listed under the "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress only ratings and functional operation of the device at these or at any other conditions above those indicated in the operational sections of this specification is not implied (while programming, follow the programming specifications).

RECOMMENDED OPERATING COND.

Commercial Devices:
Ambient Temperature (T_A) .....0 to 75°C
Supply voltage (V_{CC})
with Respect to Ground .....+4.75 to +5.25V

DC ELECTRICAL CHARACTERISTICS

Over Recommended Operating Conditions (Unless Otherwise Specified)

SYMBOL	PARAMETER	CONDITION			MIN.	TYP.[2]	MAX.	UNITS
V_{IL}	Input Low Voltage				Vss – 0.5	—	0.8	V
V_{IH}	Input High Voltage				2.0	—	Vcc+1	V
I_{IL}	Input or I/O Low Leakage Current	$0V \le V_{IN} \le V_{IL}$ (MAX.)			—	—	–10	µA
I_{IH}	Input or I/O High Leakage Current	$V_{IH} \le V_{IN} \le V_{CC}$			—	—	10	µA
V_{OL}	Output Low Voltage	I_{OL} = MAX. Vin = V_{IL} or V_{IH}			—	—	0.5	V
V_{OH}	Output High Voltage	I_{OH} = MAX. Vin = V_{IL} or V_{IH}			2.4	—	—	V
I_{OL}	Low Level Output Current				—	—	24	mA
I_{OH}	High Level Output Current				—	—	–3.2	mA
I_{OS}[1]	Output Short Circuit Current	V_{CC} = 5V V_{OUT} = 0.5V T_A = 25°C			–30	—	–150	mA
I_{CC}	Operating Power Supply Current	V_{IL} = 0.5V V_{IH} = 3.0V Outputs Open (no load)	f_{toggle} = 15MHz	L -25	—	75	90	mA
			f_{toggle} = 25MHz	L -10/-15	—	75	115	mA
			f_{toggle} = 15MHz	Q -15/-25	—	45	55	mA

1) One output at a time for a maximum duration of one second. Vout = 0.5V was selected to avoid test problems caused by tester ground degradation. Guaranteed but not 100% tested.
2) Typical values are at Vcc = 5V and T_A = 25 °C

CAPACITANCE (T_A = 25°C, f = 1.0 MHz)

SYMBOL	PARAMETER	MAXIMUM*	UNITS	TEST CONDITIONS
C_I	Input Capacitance	8	pF	V_{CC} = 5.0V, V_I = 2.0V
$C_{I/O}$	I/O Capacitance	10	pF	V_{CC} = 5.0V, $V_{I/O}$ = 2.0V

*Guaranteed but not 100% tested.

AC SWITCHING CHARACTERISTICS

Over Recommended Operating Conditions

PARAMETER	TEST COND[1].	DESCRIPTION	-10		-15		-25		UNITS
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
tpd	1	Input or I/O to Combinational Output	3	10	3	15	3	25	ns
tco	1	Clock to Output Delay	2	7	2	10	2	12	ns
tcf[2]		Clock to Feedback Delay	—	7	—	8	—	10	ns
tsu		Setup Time, Input or Feedback before Clock↑	10	—	12	—	15	—	ns
th		Hold Time, Input or Feedback after Clock↑	0	—	0	—	0	—	ns
fmax[3]	1	Maximum Clock Frequency with External Feedback, 1/(tsu + tco)	58.8	—	45.5	—	37	—	MHz
	1	Maximum Clock Frequency with Internal Feedback, 1/(tsu + tcf)	58.8	—	50	—	40	—	MHz
	1	Maximum Clock Frequency with No Feedback	62.5	—	62.5	—	41.7	—	MHz
twh[4]		Clock Pulse Duration, High	8	—	8	—	12	—	ns
twl[4]		Clock Pulse Duration, Low	8	—	8	—	12	—	ns
ten	2	Input or I/O to Output Enabled	—	10	—	15	—	25	ns
	2	OE↓ to Output Enabled	—	10	—	15	—	20	ns
tdis	3	Input or I/O to Output Disabled	—	10	—	15	—	25	ns
	3	OE↑ to Output Disabled	—	10	—	15	—	20	ns

1) Refer to **Switching Test Conditions** section.
2) Calculated from fmax with internal feedback. Refer to **fmax Descriptions** section.
3) Refer to **fmax Descriptions** section.
4) Clock pulses of widths less than the specification may be detected as valid clock signals.

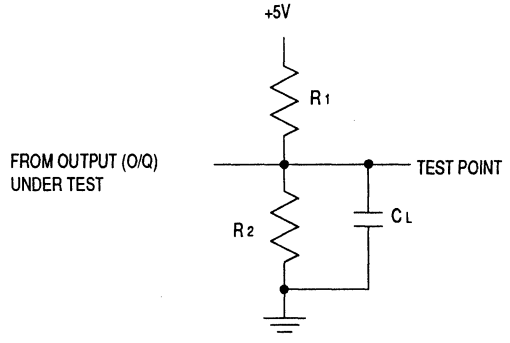
SWITCHING TEST CONDITIONS

Input Pulse Levels	GND to 3.0V
Input Rise and Fall Times	3ns 10% – 90%
Input Timing Reference Levels	1.5V
Output Timing Reference Levels	1.5V
Output Load	See Figure

3-state levels are measured 0.5V from steady-state active level.

Output Load Conditions (see figure)

Test Condition		R1	R2	CL
1		200Ω	390Ω	50pF
2	Active High	∞	390Ω	50pF
	Active Low	200Ω	390Ω	50pF
3	Active High	∞	390Ω	5pF
	Active Low	200Ω	390Ω	5pF



C L INCLUDES JIG AND PROBE TOTAL CAPACITANCE

GAL20V8A/B ORDERING INFORMATION

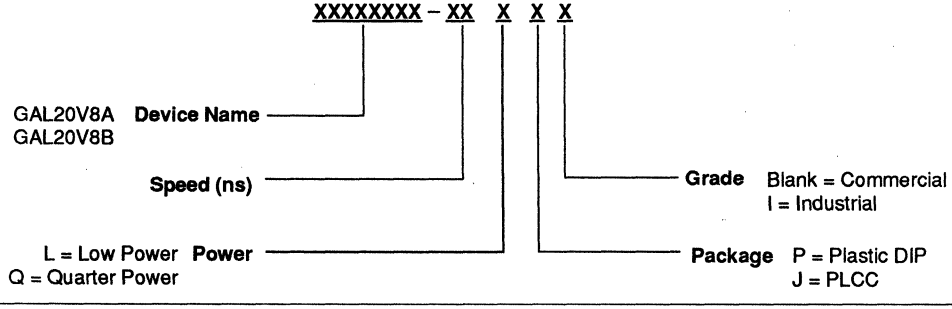
Commercial Grade Specifications

Tpd (ns)	Tsu (ns)	Tco (ns)	Icc (mA)	Ordering #	Package
7.5	7	5	115	GAL20V8B-7LP	24-Pin Plastic DIP
			115	GAL20V8B-7LJ	28-Lead PLCC
10	10	7	115	GAL20V8B-10LP	24-Pin Plastic DIP
			115	GAL20V8B-10LJ	28-Lead PLCC
			115	GAL20V8A-10LP	24-Pin Plastic DIP
			115	GAL20V8A-10LJ	28-Lead PLCC
15	12	10	55	GAL20V8A-15QP	24-Pin Plastic DIP
			55	GAL20V8A-15QJ	28-Lead PLCC
			115	GAL20V8A-15LP	24-Pin Plastic DIP
			115	GAL20V8A-15LJ	28-Lead PLCC
25	15	12	55	GAL20V8A-25QP	24-Pin Plastic DIP
			55	GAL20V8A-25QJ	28-Lead PLCC
			90	GAL20V8A-25LP	24-Pin Plastic DIP
			90	GAL20V8A-25LJ	28-Lead PLCC

Industrial Grade Specifications

Tpd (ns)	Tsu (ns)	Tco (ns)	Icc (mA)	Ordering #	Package
10	10	7	130	GAL20V8B-10LPI	24-Pin Plastic DIP
			130	GAL20V8B-10LJI	28-Lead PLCC
15	12	10	130	GAL20V8A-15LPI	24-Pin Plastic DIP
			130	GAL20V8A-15LJI	28-Lead PLCC
20	13	11	65	GAL20V8A-20QPI	24-Pin Plastic DIP
			65	GAL20V8A-20QJI	28-Lead PLCC
25	15	12	65	GAL20V8A-25QPI	24-Pin Plastic DIP
			65	GAL20V8A-25QJI	28-Lead PLCC
			130	GAL20V8A-25LPI	24-Pin Plastic DIP
			130	GAL20V8A-25LJI	28-Lead PLCC

PART NUMBER DESCRIPTION



GAL18V10

High Performance E²CMOS PLD
Generic Array Logic™

FEATURES

- **HIGH PERFORMANCE E²CMOS® TECHNOLOGY**
  - **— 15 ns Maximum Propagation Delay**
  - **— Fmax = 62.5 MHz**
  - **— 10ns Maximum from Clock Input to Data Output**
  - **— TTL Compatible 16 mA Outputs**
  - **— UltraMOS® Advanced CMOS Technology**
- **LOW POWER CMOS**
  - **— 75 mA Typical Icc**
- **ACTIVE PULL-UPS ON ALL PINS**
- **E² CELL TECHNOLOGY**
  - **— Reconfigurable Logic**
  - **— Reprogrammable Cells**
  - **— 100% Tested/Guaranteed 100% Yields**
  - **— High Speed Electrical Erasure (50ms)**
  - **— 20 Year Data Retention**
- **TEN OUTPUT LOGIC MACROCELLS**
  - **— Uses Standard 22V10 Macrocells**
  - **— Maximum Flexibility for Complex Logic Designs**
- **PRELOAD AND POWER-ON RESET OF REGISTERS**
  - **— 100% Functional Testability**
- **APPLICATIONS INCLUDE:**
  - **— DMA Control**
  - **— State Machine Control**
  - **— High Speed Graphics Processing**
  - **— Standard Logic Speed Upgrade**
- **ELECTRONIC SIGNATURE FOR IDENTIFICATION**

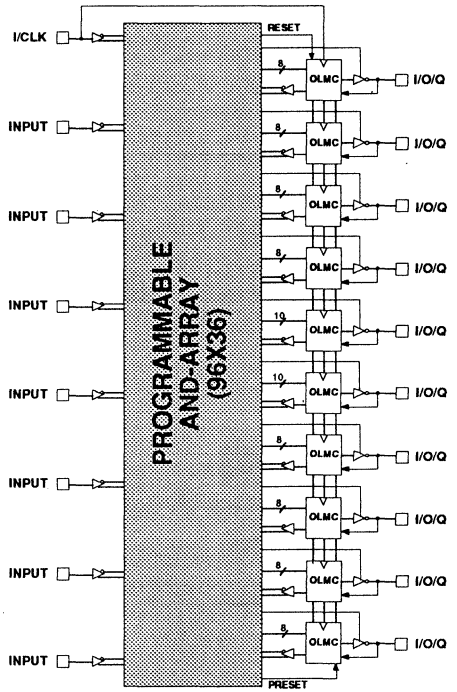
DESCRIPTION

The GAL18V10, at 15 ns maximum propagation delay time, combines a high performance CMOS process with Electrically Erasable (E²) floating gate technology to provide the highest performance 20 pin PLD available on the market. CMOS circuitry allows the GAL18V10 to consume much less power when compared to its bipolar counterparts. The E² technology offers high speed (50ms) erase times, providing the ability to reprogram or reconfigure the device quickly and efficiently.

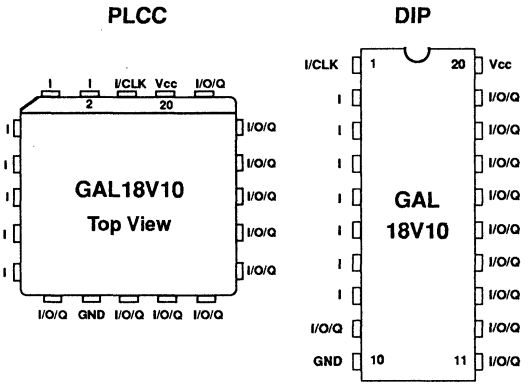
By building on the popular 22V10 architecture, the GAL18V10 allows the designer to be immediately productive, eliminating the learning curve. The generic architecture provides maximum design flexibility by allowing the Output Logic Macrocell (OLMC) to be configured by the user. The GAL18V10 OLMC is fully compatible with the OLMC in standard bipolar and CMOS 22V10 devices.

Unique test circuitry and reprogrammable cells allow complete AC, DC, and functional testing during manufacture. As a result, LATTICE is able to guarantee 100% field programmability and functionality of all GAL® products. LATTICE also guarantees 100 erase/rewrite cycles and data retention in excess of 20 years.

FUNCTIONAL BLOCK DIAGRAM



PACKAGE DIAGRAMS



LATTICE SEMICONDUCTOR CORP., 5555 N.E. Moore Ct., Hillsboro, Oregon 97124 U.S.A.
Tel. (503) 681-0118 or 1-800-FASTGAL; FAX (503) 681-3037

December 1991

ABSOLUTE MAXIMUM RATINGS(1)

Supply voltage V_{CC} ..... -0.5 to +7V
Input voltage applied ..... -2.5 to V_{CC} +1.0V
Off-state output voltage applied ..... -2.5 to V_{CC} +1.0V
Storage Temperature ..... -65 to 150°C
Ambient Temperature with
Power Applied ..... -55 to 125°C

1. Stresses above those listed under the "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress only ratings and functional operation of the device at these or at any other conditions above those indicated in the operational sections of this specification is not implied (while programming, follow the programming specifications).

RECOMMENDED OPERATING COND.

Commercial Devices:
Ambient Temperature (T_A) ..... 0 to +75°C
Supply voltage (V_{CC})
with Respect to Ground ..... +4.75 to +5.25V

DC ELECTRICAL CHARACTERISTICS

Over Recommended Operating Conditions (Unless Otherwise Specified)

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.[3]	MAX.	UNITS
V_{IL}	Input Low Voltage		Vss – 0.5	—	0.8	V
V_{IH}	Input High Voltage		2.0	—	Vcc+1	V
I_{IL}[1]	Input or I/O Low Leakage Current	$0V \le V_{IN} \le V_{IL}$ (MAX.)	—	—	–100	µA
I_{IH}	Input or I/O High Leakage Current	$3.5V \le V_{IN} \le V_{CC}$	—	—	10	µA
V_{OL}	Output Low Voltage	I_{OL} = MAX. Vin = V_{IL} or V_{IH}	—	—	0.5	V
V_{OH}	Output High Voltage	I_{OH} = MAX. Vin = V_{IL} or V_{IH}	2.4	—	—	V
I_{OL}	Low Level Output Current		—	—	16	mA
I_{OH}	High Level Output Current		—	—	–3.2	mA
I_{OS}[2]	Output Short Circuit Current	V_{CC} = 5V V_{OUT} = 0.5V T_A = 25°C	–50	—	–135	mA
I_{CC}	Operating Power Supply Current	V_{IL} = 0.5V V_{IH} = 3.0V ftoggle = 15Mhz Outputs Open	—	75	115	mA

1) The leakage current is due to the internal pull-up on all pins. See **Input Buffer** section for more information.
2) One output at a time for a maximum duration of one second. Vout = 0.5V was selected to avoid test problems caused by tester ground degradation. Guaranteed but not 100% tested.
3) Typical values are at Vcc = 5V and T_A = 25 °C

CAPACITANCE (T_A = 25°C, f = 1.0 MHz)

SYMBOL	PARAMETER	MAXIMUM*	UNITS	TEST CONDITIONS
C_I	Input Capacitance	8	pF	V_{CC} = 5.0V, V_I = 2.0V
$C_{I/O}$	I/O Capacitance	10	pF	V_{CC} = 5.0V, $V_{I/O}$ = 2.0V

*Guaranteed but not 100% tested.

AC SWITCHING CHARACTERISTICS

Over Recommended Operating Conditions

PARAMETER	TEST COND.[1]	DESCRIPTION	-15		-20		UNITS
			MIN.	MAX.	MIN.	MAX.	
tpd	1	Input or I/O to Combinatorial Output	—	15	—	20	ns
tco	1	Clock to Output Delay	—	10	—	12	ns
tcf[2]	—	Clock to Feedback Delay	—	7	—	10	ns
tsu	—	Setup Time, Input or Feedback before Clock↑	10	—	12	—	ns
th	—	Hold Time, Input or Feedback after Clock↑	0	—	0	—	ns
fmax[3]	1	Maximum Clock Frequency with External Feedback, 1/(tsu +tco)	50	—	41.6	—	MHz
	1	Maximum Clock Frequency with Internal Feedback, 1/(tsu + tcf)	58.8	—	45.4	—	MHz
	1	Maximum Clock Frequency with No Feedback	62.5	—	62.5	—	MHz
twh[4]	—	Clock Pulse Duration, High	8	—	8	—	ns
twl[4]	—	Clock Pulse Duration, Low	8	—	8	—	ns
ten	2	Input or I/O to Output Enabled	—	15	—	20	ns
tdis	3	Input or I/O to Output Disabled	—	15	—	20	ns
tar	1	Input or I/O to Asynchronous Reset of Register	—	20	—	20	ns
tarw	—	Asynchronous Reset Pulse Duration	10	—	15	—	ns
tarr	—	Asynchronous Reset to Clock↑ Recovery Time	15	—	15	—	ns
tspr	—	Synchronous Preset to Clock↑ Recovery Time	10	—	12	—	ns

1) Refer to **Switching Test Conditions** section.
2) Calculated from fmax with internal feedback. Refer to **fmax Description** section.
3) Refer to **fmax Description** section.
4) Clock pulses of widths less than the specification may be detected as valid clock signals.

GAL18V10 ORDERING INFORMATION

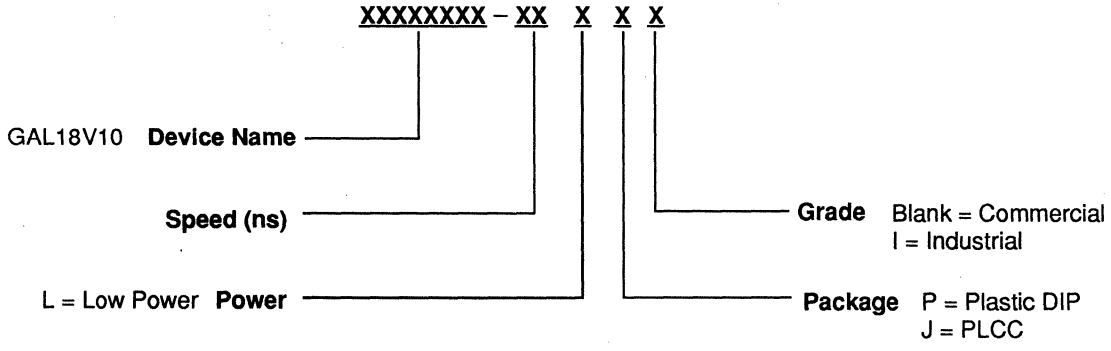
Commercial Grade Specifications

Tpd (ns)	Tsu (ns)	Tco (ns)	Icc (mA)	Ordering #	Package
15	10	10	115	GAL18V10-15LP	20-Pin Plastic DIP
			115	GAL18V10-15LJ	20-Lead PLCC
20	12	12	115	GAL18V10-20LP	20-Pin Plastic DIP
			115	GAL18V10-20LJ	20-Lead PLCC

Industrial Grade Specifications

Tpd (ns)	Tsu (ns)	Tco (ns)	Icc (mA)	Ordering #	Package
20	12	12	125	GAL18V10-20LPI	20-Pin Plastic DIP
			125	GAL18V10-20LJI	20-Lead PLCC

PART NUMBER DESCRIPTION



GAL22V10B
GAL22V10

High Performance E²CMOS PLD

FEATURES

- **HIGH PERFORMANCE E²CMOS® TECHNOLOGY**
  - **10 ns Maximum Propagation Delay**
  - **Fmax = 105 MHz**
  - **7 ns Maximum from Clock Input to Data Output**
  - **TTL Compatible 16 mA Outputs**
  - **UltraMOS® Advanced CMOS Technology**
- **ACTIVE PULL-UPS ON ALL PINS**
- **COMPATIBLE WITH STANDARD 22V10 DEVICES**
  - **Fully Function/Fuse-Map/Parametric Compatible with Bipolar and UVCMOS 22V10 Devices**
- **50% REDUCTION IN POWER VERSUS BIPOLAR**
- **E² CELL TECHNOLOGY**
  - **Reconfigurable Logic**
  - **Reprogrammable Cells**
  - **100% Tested/Guaranteed 100% Yields**
  - **High Speed Electrical Erasure (<100ms)**
  - **20 Year Data Retention**
- **TEN OUTPUT LOGIC MACROCELLS**
  - **Maximum Flexibility for Complex Logic Designs**
- **PRELOAD AND POWER-ON RESET OF REGISTERS**
  - **100% Functional Testability**
- **APPLICATIONS INCLUDE:**
  - **DMA Control**
  - **State Machine Control**
  - **High Speed Graphics Processing**
  - **Standard Logic Speed Upgrade**
- **ELECTRONIC SIGNATURE FOR IDENTIFICATION**

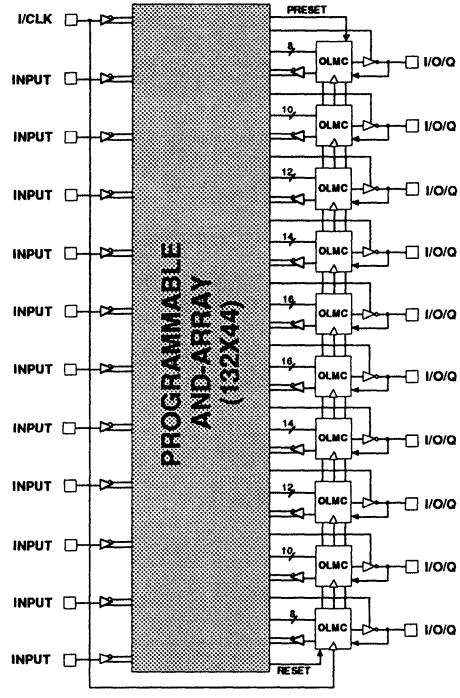
DESCRIPTION

The GAL22V10B, at 10ns maximum propagation delay time, combines a high performance CMOS process with Electrically Erasable (E^2) floating gate technology to provide the highest performance available of any 22V10 device on the market. CMOS circuitry allows the GAL22V10 to consume much less power when compared to bipolar 22V10 devices. E^2 technology offers high speed (<100ms) erase times, providing the ability to reprogram or reconfigure the device quickly and efficiently.

The generic architecture provides maximum design flexibility by allowing the Output Logic Macrocell (OLMC) to be configured by the user. The GAL22V10 is fully function/fuse map/parametric compatible with standard bipolar and CMOS 22V10 devices.

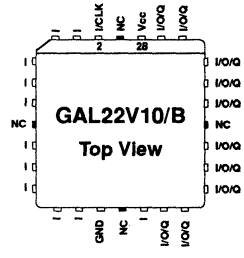
Unique test circuitry and reprogrammable cells allow complete AC, DC, and functional testing during manufacture. As a result, LATTICE is able to guarantee 100% field programmability and functionality of all GAL® products. LATTICE also guarantees 100 erase/rewrite cycles and data retention in excess of 20 years.

FUNCTIONAL BLOCK DIAGRAM

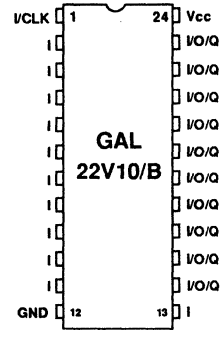


PACKAGE DIAGRAMS

PLCC



DIP



LATTICE SEMICONDUCTOR CORP., 5555 N.E. Moore Ct., Hillsboro, Oregon 97124 U.S.A.
Tel. (503) 681-0118 or 1-800-FASTGAL; FAX (503) 681-3037

December 1991

ABSOLUTE MAXIMUM RATINGS(1)

Supply voltage V_{CC} ..... -0.5 to +7V

Input voltage applied ..... -2.5 to V_{CC} +1.0V
Off-state output voltage applied ..... -2.5 to V_{CC} +1.0V
Storage Temperature ..... -65 to 150°C
Ambient Temperature with
Power Applied ..... -55 to 125°C

1. Stresses above those listed under the "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress only ratings and functional operation of the device at these or at any other conditions above those indicated in the operational sections of this specification is not implied (while programming, follow the programming specifications).

RECOMMENDED OPERATING COND.

Commercial Devices:
Ambient Temperature (T_A) ..... 0 to +75°C
Supply voltage (V_{CC})
with Respect to Ground ..... +4.75 to +5.25V

DC ELECTRICAL CHARACTERISTICS

Over Recommended Operating Conditions (Unless Otherwise Specified)

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.[3]	MAX.	UNITS
V_{IL}	Input Low Voltage		Vss – 0.5	—	0.8	V
V_{IH}	Input High Voltage		2.0	—	Vcc+1	V
I_{IL}[1]	Input or I/O Low Leakage Current	$0V \le V_{IN} \le V_{IL}$ (MAX.)	—	—	–100	µA
I_{IH}	Input or I/O High Leakage Current	$3.5V \le V_{IN} \le V_{CC}$	—	—	10	µA
V_{OL}	Output Low Voltage	I_{OL} = MAX. Vin = V_{IL} or V_{IH}	—	—	0.5	V
V_{OH}	Output High Voltage	I_{OH} = MAX. Vin = V_{IL} or V_{IH}	2.4	—	—	V
I_{OL}	Low Level Output Current		—	—	16	mA
I_{OH}	High Level Output Current		—	—	–3.2	mA
I_{OS}[2]	Output Short Circuit Current	V_{CC} = 5V V_{OUT} = 0.5V T_A = 25°C	–30	—	–130	mA
I_{CC}	Operating Power Supply Current	VIL = 0.5V VIH = 3.0V ftoggle = 25Mhz Outputs Open	—	90	130	mA

1) The leakage current is due to the internal pull-up on all pins. See **Input Buffer** section for more information.
2) One output at a time for a maximum duration of one second. Vout = 0.5V was selected to avoid test problems caused by tester ground degradation. Guaranteed but not 100% tested.
3) Typical values are at Vcc = 5V and TA = 25 °C

CAPACITANCE (T_A = 25°C, f = 1.0 MHz)

SYMBOL	PARAMETER	MAXIMUM*	UNITS	TEST CONDITIONS
C_I	Input Capacitance	8	pF	V_{CC} = 5.0V, V_I = 2.0V
$C_{I/O}$	I/O Capacitance	8	pF	V_{CC} = 5.0V, $V_{I/O}$ = 2.0V

*Guaranteed but not 100% tested.

AC SWITCHING CHARACTERISTICS

Over Recommended Operating Conditions

PARAMETER	TEST COND.[1]	DESCRIPTION	-10		-15		UNITS
			MIN.	MAX.	MIN.	MAX.	
t_{pd}	1	Input or I/O to Combinatorial Output	3	10	3	15	ns
t_{co}	1	Clock to Output Delay	2	7	2	8	ns
t_{cf}[2]	—	Clock to Feedback Delay	—	2.5	—	2.5	ns
t_{su_1}	—	Setup Time, Input or Feedback before Clock↑	7	—	10	—	ns
t_{su_2}	—	Setup Time, SP before Clock↑	10	—	10	—	ns
t_h	—	Hold Time, Input or Feedback after Clock↑	0	—	0	—	ns
f_{max}[3]	1	Maximum Clock Frequency with External Feedback, 1/(tsu + tco)	71.4	—	55.5	—	MHz
	1	Maximum Clock Frequency with Internal Feedback, 1/(tsu + tcf)	105	—	80	—	MHz
	1	Maximum Clock Frequency with No Feedback	105	—	83.3	—	MHz
t_{wh}[4]	—	Clock Pulse Duration, High	4	—	6	—	ns
t_{wl}[4]	—	Clock Pulse Duration, Low	4	—	6	—	ns
t_{en}	2	Input or I/O to Output Enabled	3	10	3	15	ns
t_{dis}	3	Input or I/O to Output Disabled	3	9	3	15	ns
t_{ar}	1	Input or I/O to Asynchronous Reset of Register	3	13	3	20	ns
t_{arw}	—	Asynchronous Reset Pulse Duration	8	—	15	—	ns
t_{arr}	—	Asynchronous Reset to Clock↑ Recovery Time	8	—	10	—	ns
t_{spr}	—	Synchronous Preset to Clock↑ Recovery Time	10	—	10	—	ns

1) Refer to **Switching Test Conditions** section.
2) Calculated from fmax with internal feedback. Refer to **fmax Description** section.
3) Refer to **fmax Description** section.
4) Clock pulses of widths less than the specification may be detected as valid clock signals.

ABSOLUTE MAXIMUM RATINGS[1]

Supply voltage V_{CC} ..... -0.5 to +7V
Input voltage applied ..... -2.5 to V_{CC} +1.0V
Off-state output voltage applied ..... -2.5 to V_{CC} +1.0V
Storage Temperature ..... -65 to 150°C
Ambient Temperature with
Power Applied ..... -55 to 125°C

1. Stresses above those listed under the "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress only ratings and functional operation of the device at these or at any other conditions above those indicated in the operational sections of this specification is not implied (while programming, follow the programming specifications).

RECOMMENDED OPERATING COND.

Commercial Devices:
Ambient Temperature (T_A) ..... 0 to +75°C
Supply voltage (V_{CC})
with Respect to Ground ..... +4.75 to +5.25V

DC ELECTRICAL CHARACTERISTICS

Over Recommended Operating Conditions (Unless Otherwise Specified)

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.[3]	MAX.	UNITS
V_{IL}	Input Low Voltage		Vss – 0.5	—	0.8	V
V_{IH}	Input High Voltage		2.0	—	Vcc+1	V
I_{IL}[1]	Input or I/O Low Leakage Current	$0V \le V_{IN} \le V_{IL}$ (MAX.)	—	—	–150	µA
I_{IH}	Input or I/O High Leakage Current	$3.5V \le V_{IN} \le V_{CC}$	—	—	10	µA
V_{OL}	Output Low Voltage	I_{OL} = MAX. Vin = V_{IL} or V_{IH}	—	—	0.5	V
V_{OH}	Output High Voltage	I_{OH} = MAX. Vin = V_{IL} or V_{IH}	2.4	—	—	V
I_{OL}	Low Level Output Current		—	—	16	mA
I_{OH}	High Level Output Current		—	—	–3.2	mA
I_{OS}[2]	Output Short Circuit Current	V_{CC} = 5V V_{OUT} = 0.5V T_A = 25°C	–50	—	–135	mA
I_{CC}	Operating Power Supply Current	VIL = 0.5V VIH = 3.0V ftoggle = 15Mhz Outputs Open	—	90	130	mA

1) The leakage current is due to the internal pull-up on all pins. See **Input Buffer** section for more information.
2) One output at a time for a maximum duration of one second. Vout = 0.5V was selected to avoid test problems caused by tester ground degradation. Guaranteed but not 100% tested.
3) Typical values are at Vcc = 5V and T_A = 25 °C

CAPACITANCE (T_A = 25°C, f = 1.0 MHz)

SYMBOL	PARAMETER	MAXIMUM*	UNITS	TEST CONDITIONS
C_I	Input Capacitance	8	pF	V_{CC} = 5.0V, V_I = 2.0V
$C_{I/O}$	I/O Capacitance	10	pF	V_{CC} = 5.0V, $V_{I/O}$ = 2.0V

*Guaranteed but not 100% tested.

AC SWITCHING CHARACTERISTICS

Over Recommended Operating Conditions

PARAMETER	TEST COND.[1]	DESCRIPTION	-15		-25		UNITS
			MIN.	MAX.	MIN.	MAX.	
t_{pd}	1	Input or I/O to Combinatorial Output	3	15	3	25	ns
t_{co}	1	Clock to Output Delay	2	8	2	15	ns
t_{cf}[2]	—	Clock to Feedback Delay	—	5	—	13	ns
t_{su}	—	Setup Time, Input or Feedback before Clock↑	12	—	15	—	ns
t_h	—	Hold Time, Input or Feedback after Clock↑	0	—	0	—	ns
f_{max}[3]	1	Maximum Clock Frequency with External Feedback, 1/(tsu + tco)	50	—	33.3	—	MHz
	1	Maximum Clock Frequency with Internal Feedback, 1/(tsu + tcf)	58.8	—	35.7	—	MHz
	1	Maximum Clock Frequency with No Feedback	62.5	—	38.5	—	MHz
t_{wh}[4]	—	Clock Pulse Duration, High	8	—	13	—	ns
t_{wl}[4]	—	Clock Pulse Duration, Low	8	—	13	—	ns
t_{en}	2	Input or I/O to Output Enabled	3	15	3	25	ns
t_{dis}	3	Input or I/O to Output Disabled	3	15	3	25	ns
t_{ar}	1	Input or I/O to Asynchronous Reset of Register	3	20	3	25	ns
t_{arw}	—	Asynchronous Reset Pulse Duration	15	—	25	—	ns
t_{arr}	—	Asynchronous Reset to Clock↑ Recovery Time	15	—	25	—	ns
t_{spr}	—	Synchronous Preset to Clock↑ Recovery Time	12	—	15	—	ns

1) Refer to **Switching Test Conditions** section.
2) Calculated from fmax with internal feedback. Refer to **fmax Description** section.
3) Refer to **fmax Description** section.
4) Clock pulses of widths less than the specification may be detected as valid clock signals.

GAL22V10/B ORDERING INFORMATION

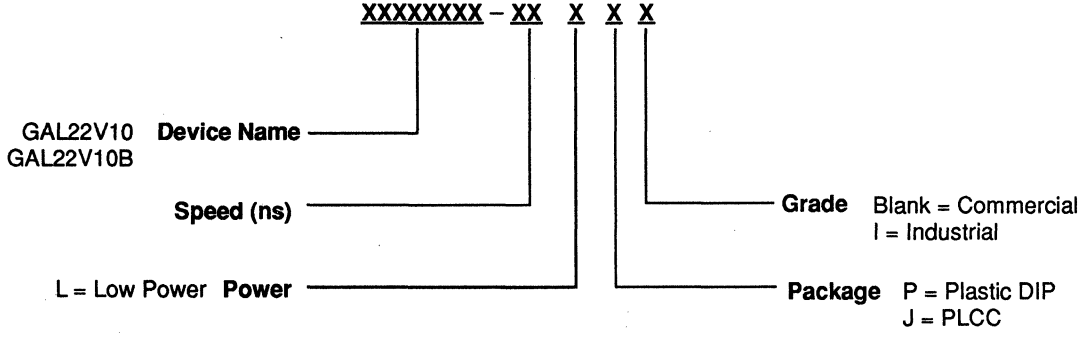
Commercial Grade Specifications

Tpd (ns)	Tsu (ns)	Tco (ns)	Icc (mA)	Ordering #	Package
10	7	7	130	GAL22V10B-10LP	24-Pin Plastic DIP
			130	GAL22V10B-10LJ	28-Lead PLCC
15	10	8	130	GAL22V10B-15LP	24-Pin Plastic DIP
			130	GAL22V10B-15LJ	28-Lead PLCC
15	12	8	130	GAL22V10-15LP	24-Pin Plastic DIP
			130	GAL22V10-15LJ	28-Lead PLCC
25	15	15	130	GAL22V10-25LP	24-Pin Plastic DIP
			130	GAL22V10-25LJ	28-Lead PLCC

Industrial Grade Specifications

Tpd (ns)	Tsu (ns)	Tco (ns)	Icc (mA)	Ordering #	Package
15	10	8	150	GAL22V10B-15LPI	24-Pin Plastic DIP
			150	GAL22V10B-15LJI	28-Lead PLCC
20	14	10	150	GAL22V10-20LPI	24-Pin Plastic DIP
			150	GAL22V10-20LJI	28-Lead PLCC
25	15	15	150	GAL22V10-25LPI	24-Pin Plastic DIP
			150	GAL22V10-25LJI	28-Lead PLCC

PART NUMBER DESCRIPTION



GAL26CV12

High Performance E²CMOS PLD
Generic Array Logic™

FEATURES

- **HIGH PERFORMANCE E²CMOS® TECHNOLOGY**
  - **15 ns Maximum Propagation Delay**
  - **Fmax = 62.5 MHz**
  - **10ns Maximum from Clock Input to Data Output**
  - **TTL Compatible 8 mA Outputs**
  - **UltraMOS® Advanced CMOS Technology**
- **ACTIVE PULL-UPS ON ALL PINS**
- **LOW POWER CMOS**
  - **90 mA Typical Icc**
- **E² CELL TECHNOLOGY**
  - **Reconfigurable Logic**
  - **Reprogrammable Cells**
  - **100% Tested/Guaranteed 100% Yields**
  - **High Speed Electrical Erasure (50ms)**
  - **20 Year Data Retention**
- **TWELVE OUTPUT LOGIC MACROCELLS**
  - **Uses Standard 22V10 Macrocells**
  - **Maximum Flexibility for Complex Logic Designs**
- **PRELOAD AND POWER-ON RESET OF REGISTERS**
  - **100% Functional Testability**
- **APPLICATIONS INCLUDE:**
  - **DMA Control**
  - **State Machine Control**
  - **High Speed Graphics Processing**
  - **Standard Logic Speed Upgrade**
- **ELECTRONIC SIGNATURE FOR IDENTIFICATION**

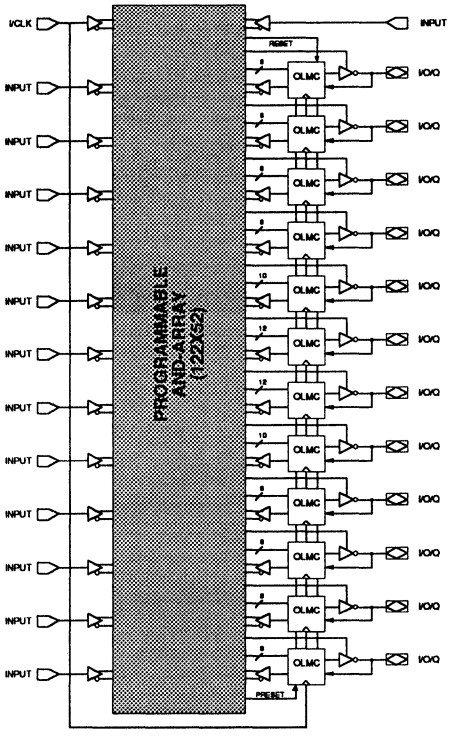
DESCRIPTION

The GAL26CV12, at 15 ns maximum propagation delay time, combines a high performance CMOS process with Electrically Erasable (E^2) floating gate technology to provide the highest performance 28 pin PLD available on the market. E^2 technology offers high speed (50ms) erase times, providing the ability to reprogram or reconfigure the device quickly and efficiently.

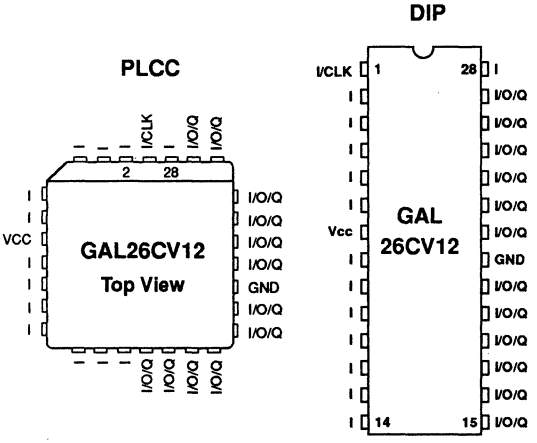
By building on the popular 22V10 architecture, the GAL26CV12 allows the designer to be immediately productive, eliminating the learning curve. The generic architecture provides maximum design flexibility by allowing the Output Logic Macrocell (OLMC) to be configured by the user. The GAL26CV12 OLMC is fully compatible with the OLMC in standard bipolar and CMOS 22V10 devices.

Unique test circuitry and reprogrammable cells allow complete AC, DC, and functional testing during manufacture. As a result, LATTICE is able to guarantee 100% field programmability and functionality of all GAL® products. LATTICE also guarantees 100 erase/rewrite cycles and data retention in excess of 20 years.

FUNCTIONAL BLOCK DIAGRAM



PACKAGE DIAGRAMS



LATTICE SEMICONDUCTOR CORP., 5555 N.E. Moore Ct., Hillsboro, Oregon 97124 U.S.A.
Tel. (503) 681-0118 or 1-800-FASTGAL; FAX (503) 681-3037

December 1991

ABSOLUTE MAXIMUM RATINGS(1)

Supply voltage V_{CC} ..... -0.5 to +7V
Input voltage applied ..... -2.5 to V_{CC} +1.0V
Off-state output voltage applied ..... -2.5 to V_{CC} +1.0V
Storage Temperature ..... -65 to 150°C
Ambient Temperature with
Power Applied ..... -55 to 125°C

1. Stresses above those listed under the "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress only ratings and functional operation of the device at these or at any other conditions above those indicated in the operational sections of this specification is not implied (while programming, follow the programming specifications).

RECOMMENDED OPERATING COND.

Commercial Devices:
Ambient Temperature (T_A) ..... 0 to +75°C
Supply voltage (V_{CC})
with Respect to Ground ..... +4.75 to +5.25V

DC ELECTRICAL CHARACTERISTICS

Over Recommended Operating Conditions (Unless Otherwise Specified)

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.[3]	MAX.	UNITS
V_{IL}	Input Low Voltage		Vss – 0.5	—	0.8	V
V_{IH}	Input High Voltage		2.0	—	Vcc+1	V
I_{IL}[1]	Input or I/O Low Leakage Current	$0V \le V_{IN} \le V_{IL}$ (MAX.)	—	—	–100	µA
I_{IH}	Input or I/O High Leakage Current	$3.5V \le V_{IN} \le V_{CC}$	—	—	10	µA
V_{OL}	Output Low Voltage	I_{OL} = MAX. Vin = V_{IL} or V_{IH}	—	—	0.5	V
V_{OH}	Output High Voltage	I_{OH} = MAX. Vin = V_{IL} or V_{IH}	2.4	—	—	V
I_{OL}	Low Level Output Current		—	—	8	mA
I_{OH}	High Level Output Current		—	—	–3.2	mA
I_{OS}[2]	Output Short Circuit Current	V_{CC} = 5V V_{OUT} = 0.5V T_A = 25°C	–50	—	–135	mA
I_{CC}	Operating Power Supply Current	V_{IL} = 0.5V V_{IH} = 3.0V ftoggle = 15Mhz Outputs Open	—	90	130	mA

1) The leakage current is due to the internal pull-up on all pins. See **Input Buffer** section for more information.
2) One output at a time for a maximum duration of one second. Vout = 0.5V was selected to avoid test problems caused by tester ground degradation. Guaranteed but not 100% tested.
3) Typical values are at Vcc = 5V and T_A = 25 °C

CAPACITANCE (T_A = 25°C, f = 1.0 MHz)

SYMBOL	PARAMETER	MAXIMUM*	UNITS	TEST CONDITIONS
C_I	Input Capacitance	8	pF	V_{CC} = 5.0V, V_I = 2.0V
$C_{I/O}$	I/O Capacitance	10	pF	V_{CC} = 5.0V, $V_{I/O}$ = 2.0V

*Guaranteed but not 100% tested.

AC SWITCHING CHARACTERISTICS

Over Recommended Operating Conditions

PARAMETER	TEST COND.[1]	DESCRIPTION	-15		-20		UNITS
			MIN.	MAX.	MIN.	MAX.	
t_{pd}	1	Input or I/O to Combinatorial Output	—	15	—	20	ns
t_{co}	1	Clock to Output Delay	—	10	—	12	ns
t_{cf}[2]	—	Clock to Feedback Delay	—	7	—	10	ns
t_{su}	—	Setup Time, Input or Feedback before Clock↑	10	—	12	—	ns
t_h	—	Hold Time, Input or Feedback after Clock↑	0	—	0	—	ns
f_{max}[3]	1	Maximum Clock Frequency with External Feedback, 1/(tsu + tco)	50	—	41.6	—	MHz
	1	Maximum Clock Frequency with Internal Feedback, 1/(tsu + tcf)	58.8	—	45.4	—	MHz
	1	Maximum Clock Frequency with No Feedback	62.5	—	62.5	—	MHz
t_{wh}[4]	—	Clock Pulse Duration, High	8	—	8	—	ns
t_{wl}[4]	—	Clock Pulse Duration, Low	8	—	8	—	ns
t_{en}	2	Input or I/O to Output Enabled	—	15	—	20	ns
t_{dis}	3	Input or I/O to Output Disabled	—	15	—	20	ns
t_{ar}	1	Input or I/O to Asynchronous Reset of Register	—	20	—	20	ns
t_{arw}	—	Asynchronous Reset Pulse Duration	10	—	15	—	ns
t_{arr}	—	Asynchronous Reset to Clock↑ Recovery Time	15	—	15	—	ns
t_{spr}	—	Synchronous Preset to Clock↑ Recovery Time	10	—	12	—	ns

1) Refer to **Switching Test Conditions** section.
2) Calculated from fmax with internal feedback. Refer to **fmax Description** section.
3) Refer to **fmax Description** section.
4) Clock pulses of widths less than the specification may be detected as valid clock signals.

GAL26CV12 ORDERING INFORMATION

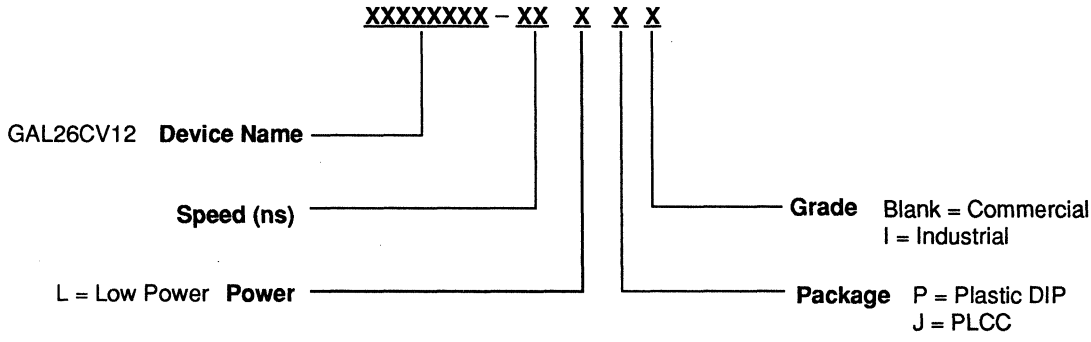
Commercial Grade Specifications

Tpd (ns)	Tsu (ns)	Tco (ns)	Icc (mA)	Ordering #	Package
15	10	10	130	GAL26CV12-15LP	28-Pin Plastic DIP
			130	GAL26CV12-15LJ	28-Lead PLCC
20	12	12	130	GAL26CV12-20LP	28-Pin Plastic DIP
			130	GAL26CV12-20LJ	28-Lead PLCC

Industrial Grade Specifications

Tpd (ns)	Tsu (ns)	Tco (ns)	Icc (mA)	Ordering #	Package
20	12	12	150	GAL26CV12-20LPI	28-Pin Plastic DIP
			150	GAL26CV12-20LJI	28-Lead PLCC

PART NUMBER DESCRIPTION



GAL20RA10

High-Speed Asynchronous E²CMOS PLD
Generic Array Logic™

FEATURES

- **HIGH PERFORMANCE E²CMOS® TECHNOLOGY**
  - **12 ns Maximum Propagation Delay**
  - **Fmax = 71.4 MHz**
  - **12 ns Maximum from Clock Input to Data Output**
  - **TTL Compatible 8 mA Outputs**
  - **UltraMOS® Advanced CMOS Technology**
- **50% to 75% REDUCTION IN POWER FROM BIPOLAR**
  - **75mA Typ I_{cc}**
- **ACTIVE PULL-UPS ON ALL PINS**
- **E² CELL TECHNOLOGY**
  - **Reconfigurable Logic**
  - **Reprogrammable Cells**
  - **100% Tested/Guaranteed 100% Yields**
  - **High Speed Electrical Erasure (<50 ms)**
  - **20 Year Data Retention**
- **TEN OUTPUT LOGIC MACROCELLS**
  - **Independent Programmable Clocks**
  - **Independent Asynchronous Reset and Preset**
  - **Registered or Combinatorial with Polarity**
  - **Full Function and Parametric Compatibility with PAL20RA10**
- **PRELOAD AND POWER-ON RESET OF ALL REGISTERS**
  - **100% Functional Testability**
- **APPLICATIONS INCLUDE:**
  - **State Machine Control**
  - **Standard Logic Consolidation**
  - **Multiple Clock Logic Designs**
- **ELECTRONIC SIGNATURE FOR IDENTIFICATION**

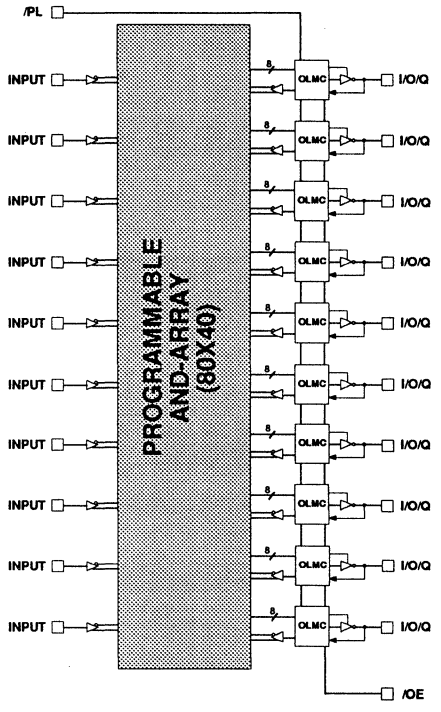
DESCRIPTION

The GAL20RA10 combines a high performance CMOS process with electrically erasable (E^2) floating gate technology to provide the highest speed performance available in the PLD market. Lattice's E²CMOS circuitry achieves power levels as low as 75mA typical I_{cc} which represents a substantial savings in power when compared to bipolar counterparts. E² technology offers high speed (<50ms) erase times providing the ability to reprogram, reconfigure or test the devices quickly and efficiently.

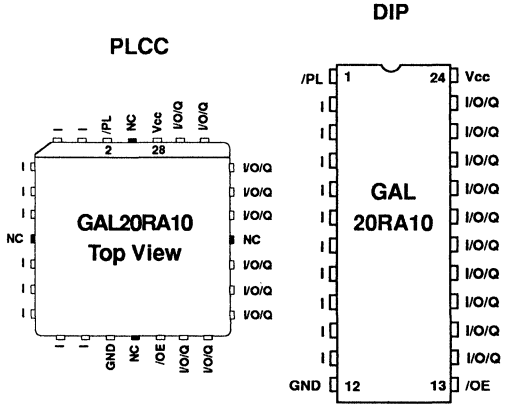
The generic architecture provides maximum design flexibility by allowing the Output Logic Macrocell (OLMC) to be configured by the user. The GAL20RA10 is a direct parametric compatible CMOS replacement for the PAL20RA10 device.

Unique test circuitry and reprogrammable cells allow complete AC,DC, and functional testing during manufacturing. Therefore, LATTICE guarantees 100% field programmability and functionality of all GAL products. LATTICE also guarantees 100 erase/rewrite cycles and that data retention exceeds 20 years.

FUNCTIONAL BLOCK DIAGRAM



PIN CONFIGURATION



LATTICE SEMICONDUCTOR CORP., 5555 N.E. Moore Ct., Hillsboro, Oregon 97124, U.S.A.
Tel. (503) 681-0118; 1-800-FASTGAL; FAX (503) 681-3037

December 1991

ABSOLUTE MAXIMUM RATINGS(1)

Supply voltage V_{CC} .....–0.5 to +7V
Input voltage applied .....–2.5 to V_{CC} +1.0V
Off-state output voltage applied .....–2.5 to V_{CC} +1.0V
Storage Temperature .....–65 to 150°C
Ambient Temperature with
Power Applied .....-55 to 125°C

1. Stresses above those listed under the "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress only ratings and functional operation of the device at these or at any other conditions above those indicated in the operational sections of this specification is not implied (while programming, follow the programming specifications).

RECOMMENDED OPERATING COND.

Commercial Devices:
Ambient Temperature (T_A) .....0 to +75°C
Supply voltage (V_{CC})
with Respect to Ground .....+4.75 to +5.25V

DC ELECTRICAL CHARACTERISTICS

Over Recommended Operating Conditions (Unless Otherwise Specified)

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.[3]	MAX.	UNITS
V_{IL}	Input Low Voltage		Vss – 0.5	—	0.8	V
V_{IH}	Input High Voltage		2.0	—	Vcc+1	V
I_{IL}[1]	Input or I/O Low Leakage Current	$0V \le V_{IN} \le V_{IL}$ (MAX.)	—	—	–100	µA
I_{IH}	Input or I/O High Leakage Current	$3.5V \le V_{IN} \le V_{CC}$	—	—	10	µA
V_{OL}	Output Low Voltage	I_{OL} = MAX. Vin = V_{IL} or V_{IH}	—	—	0.5	V
V_{OH}	Output High Voltage	I_{OH} = MAX. Vin = V_{IL} or V_{IH}	2.4	—	—	V
I_{OL}	Low Level Output Current		—	—	8	mA
I_{OH}	High Level Output Current		—	—	–3.2	mA
I_{OS}[2]	Output Short Circuit Current	V_{CC} = 5V V_{OUT} = 0.5V T_A = 25°C	–50	—	–135	mA
ICC	Operating Power Supply Current	VIL = 0.5V VIH = 3.0V ftoggle = 15Mhz Outputs Open	—	75	100	mA

1) The leakage current is due to the internal pull-up resistor on all pins. See **Input Buffer** section for more information.
2) One output at a time for a maximum duration of one second. Vout = 0.5V was selected to avoid test problems caused by tester ground degradation. Guaranteed but not 100% tested.
3) Typical values are at Vcc = 5V and TA = 25 °C

CAPACITANCE (T_A = 25°C, f = 1.0 MHz)

SYMBOL	PARAMETER	MAXIMUM*	UNITS	TEST CONDITIONS
C_I	Input Capacitance	8	pF	V_{CC} = 5.0V, V_I = 2.0V
$C_{I/O}$	I/O Capacitance	10	pF	V_{CC} = 5.0V, $V_{I/O}$ = 2.0V

*Guaranteed but not 100% tested.

AC SWITCHING CHARACTERISTICS

Over Recommended Operating Conditions

PARAMETER	TEST COND.[1]	DESCRIPTION	-12		-15		-20		-30		UNITS
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
tpd	1	Input or I/O to Combinatorial Output	—	12	—	15	—	20	—	30	ns
tco	1	Clock to Output Delay	—	12	—	15	—	20	—	30	ns
tsu	—	Setup Time, Input or Feedback before Clock	4	—	7	—	10	—	20	—	ns
th	—	Hold Time, Input or Feedback after Clock	3	—	3	—	3	—	10	—	ns
fmax[2]	1	Maximum Clock Frequency with External Feedback, 1/(tsu + tco)	62.5	—	45.0	—	33.3	—	20.0	—	MHz
	1	Maximum Clock Frequency without Feedback	71.4	—	50.0	—	41.7	—	25.0	—	MHz
twh[3]	—	Clock Pulse Duration, High	7	—	10	—	12	—	20	—	ns
twl[3]	—	Clock Pulse Duration, Low	7	—	10	—	12	—	20	—	ns
ten / tdis	2,3	Input or I/O to Output Enabled / Disabled	—	12	—	15	—	20	—	30	ns
ten / tdis	2,3	$\overline{OE}$ to Output Enabled / Disabled	—	9	—	12	—	15	—	20	ns
tar / tap	1	Input or I/O to Asynchronous Reset / Preset	—	12	—	15	—	20	—	30	ns
tarw / tapw	—	Asynchronous Reset / Preset Pulse Duration	12	—	15	—	20	—	20	—	ns
tarr / tapr	—	Asynchronous Reset / Preset Recovery Time	7	—	10	—	12	—	20	—	ns
twp	—	Preload Pulse Duration	12	—	15	—	20	—	30	—	ns
tsp	—	Preload Setup Time	7	—	10	—	15	—	25	—	ns
thp	—	Preload Hold Time	7	—	10	—	15	—	25	—	ns

1) Refer to **Switching Test Conditions** section.
2) Refer to **fmax Descriptions** section.
3) Clock pulses of widths less than the specification may be detected as valid clock signals.

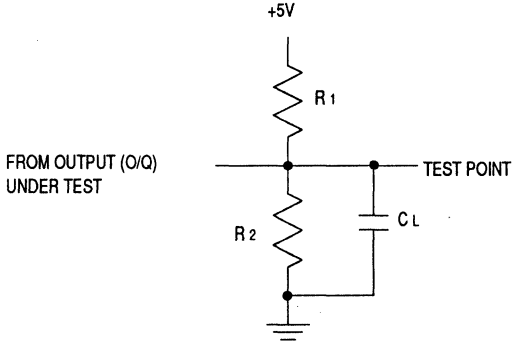
SWITCHING TEST CONDITIONS

Input Pulse Levels	GND to 3.0V
Input Rise and Fall Times	3ns 10% – 90%
Input Timing Reference Levels	1.5V
Output Timing Reference Levels	1.5V
Output Load	See Figure

3-state levels are measured 0.5V from steady-state active level.

Output Load Conditions (see figure)

Test Condition		R1	R2	CL
1		470Ω	390Ω	50pF
2	Active High	∞	390Ω	50pF
	Active Low	470Ω	390Ω	50pF
3	Active High	∞	390Ω	5pF
	Active Low	470Ω	390Ω	5pF



C L INCLUDES JIG AND PROBE TOTAL CAPACITANCE

GAL20RA10 ORDERING INFORMATION

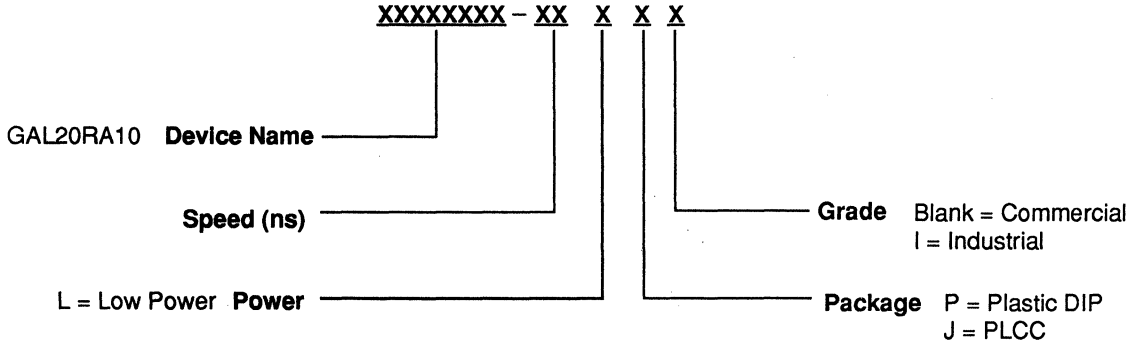
Commercial Grade Specifications

Tpd (ns)	Tsu (ns)	Tco (ns)	Icc (mA)	Ordering #	Package
12	4	12	100	GAL20RA10-12LP	24-Pin Plastic DIP
			100	GAL20RA10-12LJ	28-Lead PLCC
15	7	15	100	GAL20RA10-15LP	24-Pin Plastic DIP
			100	GAL20RA10-15LJ	28-Lead PLCC
20	10	20	100	GAL20RA10-20LP	24-Pin Plastic DIP
			100	GAL20RA10-20LJ	28-Lead PLCC
30	20	30	100	GAL20RA10-30LP	24-Pin Plastic DIP
			100	GAL20RA10-30LJ	28-Lead PLCC

Industrial Grade Specifications

Tpd (ns)	Tsu (ns)	Tco (ns)	Icc (mA)	Ordering #	Package
20	10	20	120	GAL20RA10-20LPI	24-Pin Plastic DIP
			120	GAL20RA10-20LJI	28-Lead PLCC

PART NUMBER DESCRIPTION



GAL20XV10B

High-Speed E²CMOS PLD
Generic Array Logic™

FEATURES

- **HIGH PERFORMANCE E²CMOS® TECHNOLOGY**
  - **— 10 ns Maximum Propagation Delay**
  - **— Fmax = 100 MHz**
  - **— 7 ns Maximum from Clock Input to Data Output**
  - **— TTL Compatible 16 mA Outputs**
  - **— UltraMOS® Advanced CMOS Technology**
- **50% to 75% REDUCTION IN POWER FROM BIPOLAR**
  - **— 75mA Typ Icc**
- **ACTIVE PULL-UPS ON ALL PINS**
- **E² CELL TECHNOLOGY**
  - **— Reconfigurable Logic**
  - **— Reprogrammable Cells**
  - **— 100% Tested/Guaranteed 100% Yields**
  - **— High Speed Electrical Erasure (<100 ms)**
  - **— 20 Year Data Retention**
- **TEN OUTPUT LOGIC MACROCELLS**
  - **— XOR Gate Capability on all Outputs**
  - **— Full Function and Parametric Compatibility with PAL12L10, 20L10, 20X10, 20X8, 20X4**
  - **— Registered or Combinatorial with Polarity**
- **PRELOAD AND POWER-ON RESET OF ALL REGISTERS**
- **APPLICATIONS INCLUDE:**
  - **— High Speed Counters**
  - **— Graphics Processing**
  - **— Comparators**
- **ELECTRONIC SIGNATURE FOR IDENTIFICATION**

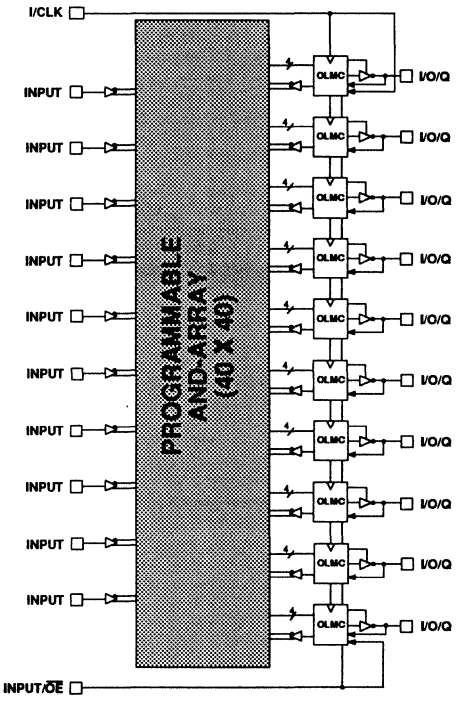
DESCRIPTION

The GAL20XV10B combines a high performance CMOS process with electrically erasable (E^2) floating gate technology to provide the highest speed Exclusive-OR PLD available in the market. At 90mA maximum Icc (75mA typical Icc), the GAL20XV10B will have a substantial savings in power when compared to bipolar counterparts. E²CMOS technology offers high speed (<100ms) erase times providing the ability to reprogram, reconfigure or test the devices quickly and efficiently.

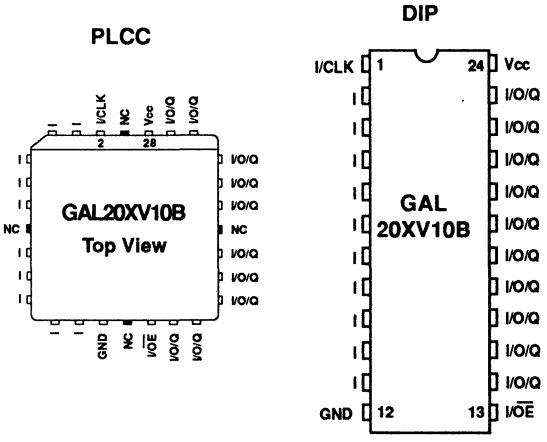
The generic architecture provides maximum design flexibility by allowing the Output Logic Macrocell (OLMC) to be configured by the user. An important subset of the many architecture configurations possible with the GAL20XV10B are the PAL® architectures listed in the macrocell description section of this document. The GAL20XV10B is capable of emulating these PAL architectures with full function and parametric compatibility.

Unique test circuitry and reprogrammable cells allow complete AC, DC, and functional testing during manufacturing. As a result, LATTICE is able to guarantee 100% field programmability and functionality of all GAL® products. LATTICE also guarantees 100 erase/rewrite cycles and data retention in excess of 20 years.

FUNCTIONAL BLOCK DIAGRAM



PACKAGE DIAGRAMS



LATTICE SEMICONDUCTOR CORP., 5555 N.E. Moore Ct., Hillsboro, Oregon 97124, U.S.A.
Tel. (503) 681-0118; 1-800-FASTGAL; FAX (503) 681-3037

December 1991

ABSOLUTE MAXIMUM RATINGS[1]

Supply voltage Vcc .....–0.5 to+7V
Input voltage applied .....–2.5 to Vcc +1.0V
Off-state output voltage applied .....–2.5 to Vcc +1.0V
Storage Temperature .....–65 to 150°C
Ambient Temperature with
Power Applied .....–55 to 125°C

1. Stresses above those listed under the "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress only ratings and functional operation of the device at these or at any other conditions above those indicated in the operational sections of this specification is not implied (while programming, follow the programming specifications).

RECOMMENDED OPERATING COND.

Commercial Devices:
Ambient Temperature (TA) .....0 to +75°C
Supply voltage (Vcc)
with Respect to Ground .....+4.75 to +5.25V

DC ELECTRICAL CHARACTERISTICS

Over Recommended Operating Conditions (Unless Otherwise Specified)

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.[3]	MAX.	UNITS
V_{IL}	Input Low Voltage		$V_{SS}-0.5$	—	0.8	V
V_{IH}	Input High Voltage		2.0	—	$V_{CC}+1$	V
I_{IL}[1]	Input or I/O Low Leakage Current	$0V \le V_{IN} \le V_{IL}$ (MAX.)	—	—	–100	µA
I_{IH}	Input or I/O High Leakage Current	$3.5V \le V_{IN} \le V_{CC}$	—	—	10	µA
V_{OL}	Output Low Voltage	I_{OL} = MAX. Vin = V_{IL} or V_{IH}	—	—	0.5	V
V_{OH}	Output High Voltage	I_{OH} = MAX. Vin = V_{IL} or V_{IH}	2.4	—	—	V
I_{OL}	Low Level Output Current		—	—	16	mA
I_{OH}	High Level Output Current		—	—	–3.2	mA
I_{OS}[2]	Output Short Circuit Current	V_{CC} = 5V V_{OUT} = 0.5V T_A= 25°C	–50	—	–150	mA
I_{CC}	Operating Power Supply Current	V_{IL} = 0.5V V_{IL} = 3.0V ftoggle = 25Mhz Outputs Open	—	75	90	mA

1) The leakage current is due to the internal pull-up on all input and I/O pins. See **Input Buffer** section for more information.
2) One output at a time for a maximum duration of one second. Vout = 0.5V was selected to avoid test problems by tester ground degradation. Guaranteed but not 100% tested.
3) Typical values are at V_{CC} = 5V and T_A = 25 °C

CAPACITANCE (T_A = 25 C, f = 1.0 MHz)

SYMBOL	PARAMETER	MAXIMUM*	UNITS	TEST CONDITIONS
C_I	Input Capacitance	8	pF	V_{CC} = 5.0V, V_I = 2.0V
$C_{I/O}$	I/O Capacitance	8	pF	V_{CC} = 5.0V, $V_{I/O}$ = 2.0V

*Guaranteed but not 100% tested.

AC SWITCHING CHARACTERISTICS

Over Recommended Operating Conditions

PARAMETER	TEST COND.[1]	DESCRIPTION	-10		-15		UNITS
			MIN.	MAX.	MIN.	MAX.	
t_{pd}	1	Input or I/O to Combinatorial Output	3	10	3	15	ns
t_{co}	1	Clock to Output Delay	2	7	2	8	ns
t_{cf}[2]	—	Clock to Feedback Delay	—	3	—	3	ns
t_{su}		Setup Time, Input or Feedback before Clock	7	—	9	—	ns
t_h		Hold Time, Input or Feedback after Clock	0	—	0	—	ns
f_{max}[3]	1	Maximum Clock Frequency with External Feedback, 1/(tsu + tco)	71.4	—	58.8	—	MHz
	1	Maximum Clock Frequency with Internal Feedback, 1/(tsu + tcf)	100	—	83.3	—	MHz
	1	Maximum Clock Frequency with No Feedback	100	—	83.3	—	MHz
t_{wh}[4]		Clock Pulse Duration, High	4	—	6	—	ns
t_{wl}[4]		Clock Pulse Duration, Low	4	—	6	—	ns
t_{en}	2	Input or I/O to Output Enabled	3	10	3	15	ns
	2	$\overline{OE}$ to Output Enabled	2	9	2	10	ns
t_{dis}	3	Input or I/O to Output Disabled	3	9	3	15	ns
	3	$\overline{OE}$ to Output Disabled	2	9	2	10	ns

1) Refer to **Switching Test Conditions** section.
2) Calculated from fmax with internal feedback. Refer to **fmax Description** section.
3) Refer to **fmax Description** section.
4) Clock pulses of widths less than the specification may be detected as valid clock signals.

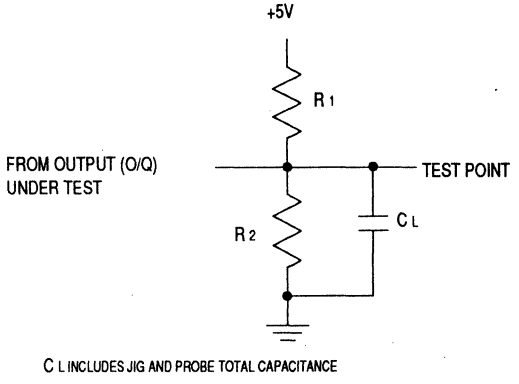
SWITCHING TEST CONDITIONS

Input Pulse Levels	GND to 3.0V
Input Rise and Fall Times	3ns 10% – 90%
Input Timing Reference Levels	1.5V
Output Timing Reference Levels	1.5V
Output Load	See Figure

3-state levels are measured 0.5V from steady-state active level.

Output Load Conditions (see figure)

Test Condition		R_1	R_2	C_L
1		300Ω	390Ω	50pF
2	Active High	∞	390Ω	50pF
	Active Low	300Ω	390Ω	50pF
3	Active High	∞	390Ω	5pF
	Active Low	300Ω	390Ω	5pF



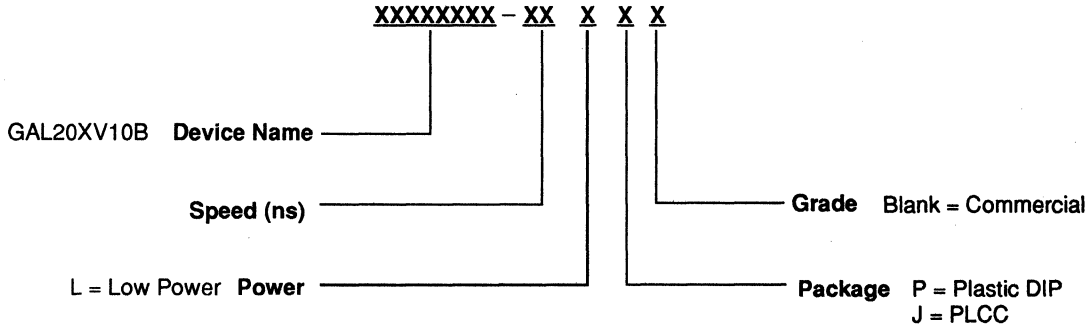
C L INCLUDES JIG AND PROBE TOTAL CAPACITANCE

GAL20XV10B ORDERING INFORMATION

Commercial Grade Specifications

Tpd (ns)	Tsu (ns)	Tco (ns)	Icc (mA)	Ordering #	Package
10	6	7	90	GAL20XV10B-10LP	24-Pin Plastic DIP
				GAL20XV10B-10LJ	28-Lead PLCC
15	8	8	90	GAL20XV10B-15LP	24-Pin Plastic DIP
				GAL20XV10B-15LJ	28-Lead PLCC

PART NUMBER DESCRIPTION



GAL6001

High Performance E²CMOS FPLA Generic Array Logic™

FEATURES

- **ELECTRICALLY ERASABLE CELL TECHNOLOGY**
  - **— Instantly Reconfigurable Logic**
  - **— Instantly Reprogrammable Cells**
  - **— Guaranteed 100% Yields**
- **HIGH PERFORMANCE E²CMOS® TECHNOLOGY**
  - **— Low Power: 90mA Typical**
  - **— High Speed: 12ns Max. Clock to Output Delay**
    25ns Min. Setup Time
    30ns Max. Propagation Delay
- **UNPRECEDENTED FUNCTIONAL DENSITY**
  - **— 78 x 64 x 36 FPLA Architecture**
  - **— 10 Output Logic Macrocells**
  - **— 8 Buried Logic Macrocells**
  - **— 20 Input and I/O Logic Macrocells**
- **HIGH-LEVEL DESIGN FLEXIBILITY**
  - **— Asynchronous or Synchronous Clocking**
  - **— Separate State Register and Input Clock Pins**
  - **— Functionally Supersets Existing 24-pin PAL® and IFL™ Devices**
- **TTL COMPATIBLE INPUTS AND OUTPUTS**
- **SPACE SAVING 24-PIN, 300-MIL DIP**
- **HIGH SPEED PROGRAMMING ALGORITHM**
- **APPLICATIONS INCLUDE:**
  - **— Sequencer**
  - **— State Machine Control**
  - **— Multiple PLD Device Integration**

DESCRIPTION

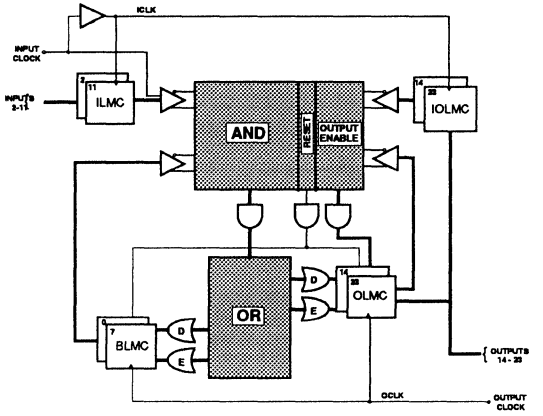
Using a high performance E²CMOS technology, Lattice Semiconductor has produced a next-generation programmable logic device, the GAL6001. Having an FPLA architecture, known for its superior flexibility in state-machine design, the GAL6001 offers the highest degree of functional integration, flexibility, and speed currently available in a 24-pin, 300-mil package.

The GAL6001 has 10 programmable Output Logic Macrocells (OLMC) and 8 programmable Buried Logic Macrocells (BLMC). In addition, there are 10 Input Logic Macrocells (ILMC) and 10 I/O Logic Macrocells (IOLMC). Two clock inputs are provided for independent control of the input and output macrocells.

Advanced features that simplify programming and reduce test time, coupled with E²CMOS reprogrammable cells, enable 100% AC, DC, programmability, and functionality testing of each GAL6001 during manufacture. This allows Lattice to guarantee 100% performance to specifications. In addition, data retention of 20 years and a minimum of 100 erase/write cycles are guaranteed.

Programming is accomplished using standard hardware and software tools. In addition, an Electronic Signature is available for storage of user specified data, and a security cell is provided to protect proprietary designs.

FUNCTIONAL BLOCK DIAGRAM



MACROCELL NAMES

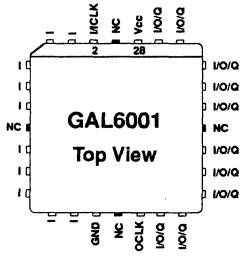
ILMC	INPUT LOGIC MACROCELL
IOLMC	I/O LOGIC MACROCELL
BLMC	BURIED LOGIC MACROCELL
OLMC	OUTPUT LOGIC MACROCELL

PIN NAMES

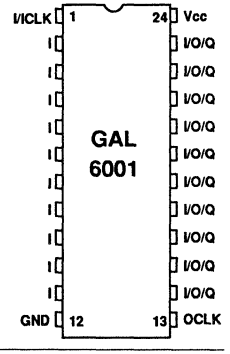
I_0 - I_{10}	INPUT	I/O/Q	BIDIRECTIONAL
ICLK	INPUT CLOCK	V_{CC}	POWER (+5)
OCLK	OUTPUT CLOCK	GND	GROUND

PIN CONFIGURATION

PLCC



DIP



LATTICE SEMICONDUCTOR CORP., 5555 N.E. Moore Ct., Hillsboro, Oregon U.S.A.
Tel. (503) 681-0118 or 1-800-FAST GAL; FAX (503) 681-3037

December 1991

ABSOLUTE MAXIMUM RATINGS(1)

Supply voltage V_{CC} .....–0.5 to +7V
Input voltage applied .....–2.5 to V_{CC} +1.0V
Off-state output voltage applied .....–2.5 to V_{CC} +1.0V
Storage Temperature .....–65 to 150°C
Ambient Temperature with
Power Applied .....–55 to 125°C

1. Stresses above those listed under the "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress only ratings and functional operation of the device at these or at any other conditions above those indicated in the operational sections of this specification is not implied (while programming, follow the programming specifications).

RECOMMENDED OPERATING COND.

Commercial Devices:
Ambient Temperature (T_A) .....0 to 75°C
Supply voltage (V_{CC})
with Respect to Ground .....+4.75 to +5.25V

DC ELECTRICAL CHARACTERISTICS

Over Recommended Operating Conditions (Unless Otherwise Specified)

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.[2]	MAX.	UNITS
V_{IL}	Input Low Voltage		Vss – 0.5	—	0.8	V
V_{IH}	Input High Voltage		2.0	—	Vcc+1	V
I_{IL}	Input or I/O Low Leakage Current	$0V \le V_{IN} \le V_{IL}$ (MAX.)	—	—	-10	µA
I_{IH}	Input or I/O High Leakage Current	$V_{IH} \le V_{IN} \le V_{CC}$	—	—	10	µA
V_{OL}	Output Low Voltage	I_{OL} = MAX. Vin = V_{IL} or V_{IH}	—	—	0.5	V
V_{OH}	Output High Voltage	I_{OH} = MAX. Vin = V_{IL} or V_{IH}	2.4	—	—	V
I_{OL}	Low Level Output Current		—	—	16	mA
I_{OH}	High Level Output Current		—	—	–3.2	mA
I_{OS}[1]	Output Short Circuit Current	V_{CC} = 5V V_{OUT} = 0.5V	–30	—	–130	mA
I_{CC}	Operating Power Supply Current	V_{IL} = 0.5V V_{IH} = 3.0V ftoggle = 15MHz Outputs Open (no load)	—	90	150	mA

1) One output at a time for a maximum duration of one second. Vout = 0.5V was selected to avoid test problems caused by tester ground degradation. Guaranteed but not 100% tested.
2) Typical values are at Vcc = 5V and T_A = 25 °C

CAPACITANCE (T_A = 25°C, f = 1.0 MHz)

SYMBOL	PARAMETER	MAXIMUM*	UNITS	TEST CONDITIONS
C_I	Input Capacitance	8	pF	V_{CC} = 5.0V, V_I = 2.0V
$C_{I/O}$	I/O Capacitance	10	pF	V_{CC} = 5.0V, $V_{I/O}$ = 2.0V

*Guaranteed but not 100% tested.

AC SWITCHING CHARACTERISTICS

Over Recommended Operating Conditions

PARAMETER	TEST COND[1].	DESCRIPTION	-30		-35		UNITS
			MIN.	MAX.	MIN.	MAX.	
t_{pd1}	1	Combinatorial Input to Combinatorial Output	—	30	—	35	ns
t_{pd2}	1	Feedback or I/O to Combinational Output	—	30	—	35	ns
t_{pd3}	1	Transparent Latch Input to Combinatorial Output	—	35	—	40	ns
t_{co1}	1	Input Latch ICLK↑ to Combinatorial Output Delay	—	35	—	40	ns
t_{co2}	1	Input Reg. ICLK↑ to Combinatorial Output Delay	—	35	—	40	ns
t_{co3}	1	Output D/E Reg. OCLK↑ to Output Delay	—	12	—	13.5	ns
t_{co4}	1	Output D Reg. Sum Term CLK↑ to Output Delay	—	35	—	40	ns
t_{su1}	—	Setup Time, Input before Input Latch ICLK↓	2.5	—	3.5	—	ns
t_{su2}	—	Setup Time, Input before Input Reg. ICLK↑	2.5	—	3.5	—	ns
t_{su3}	—	Setup Time, Input or Feedback before D/E Reg. OCLK↑	25	—	30	—	ns
t_{su4}	—	Setup Time, Input or Feedback before D Reg. Sum Term CLK↑	7.5	—	10	—	ns
t_{su5}	—	Setup Time, Input Reg. ICLK↑ before D/E Reg. OCLK↑	30	—	35	—	ns
t_{su6}	—	Setup Time, Input Reg. ICLK↑ before D Reg. Sum Term CLK↑	15	—	17	—	ns
t_{h1}	—	Hold Time, Input after Input Latch ICLK↓	5	—	5	—	ns
t_{h2}	—	Hold Time, Input after Input Reg. ICLK↑	5	—	5	—	ns
t_{h3}	—	Hold Time, Input or Feedback after D/E Reg. OCLK↑	-5	—	-5	—	ns
t_{h4}	—	Hold Time, Input or Feedback after D Reg. Sum Term CLK↑	10	—	12.5	—	ns
f_{max}	—	Maximum Clock Frequency, OCLK	27	—	22.9	—	MHz
t_{wh1}[2]	—	ICLK or OCLK Pulse Duration, High	10	—	10	—	ns
t_{wh2}[2]	—	Sum Term CLK Pulse Duration, High	15	—	15	—	ns
t_{wl1}[2]	—	ICLK or OCLK Pulse Duration, Low	10	—	10	—	ns
t_{wl2}[2]	—	Sum Term CLK Pulse Duration, Low	15	—	15	—	ns
t_{arw}	—	Reset Pulse Duration	15	—	15	—	ns
t_{en}	2	Input or I/O to Output Enabled	—	25	—	30	ns
t_{dis}	3	Input or I/O to Output Disabled	—	25	—	30	ns
t_{ar}	1	Input or I/O to Asynchronous Reg. Reset	—	35	—	35	ns
t_{arr1}	—	Asynchronous Reset to OCLK Recovery Time	20	—	20	—	ns
t_{arr2}	—	Asynchronous Reset to Sum Term CLK Recovery Time	10	—	10	—	ns

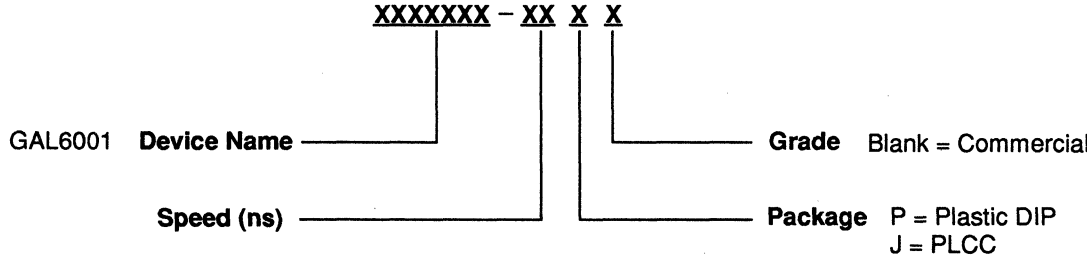
1) Refer to **Switching Test Conditions** section.
2) Clock pulses of widths less than the specification may be detected as valid clock signals.

GAL6001 ORDERING INFORMATION

Commercial Grade Specifications

Tpd (ns)	Fclk (MHz)	Icc (mA)	Ordering #	Package
30	27	150	GAL6001-30P	24-Pin Plastic DIP
		150	GAL6001-30J	28-Lead PLCC
35	22.9	150	GAL6001-35P	24-Pin Plastic DIP
		150	GAL6001-35J	28-Lead PLCC

GAL6001 ORDERING INFORMATION



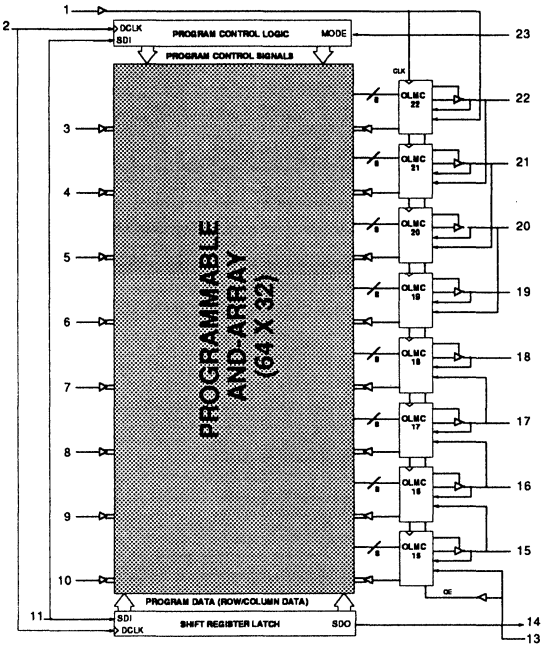
ispGAL16Z8

In-System Programmable High Performance E²CMOS PLD

FEATURES

- **IN-SYSTEM-PROGRAMMABLE — 5-VOLT ONLY**
  - **— Change Logic "On The Fly" in Seconds**
  - **— Non-volatile E² Technology**
- **MINIMUM 10,000 ERASE/WRITE CYCLES**
- **DIAGNOSTIC MODE FOR CONTROLLING AND OBSERVING SYSTEM LOGIC**
- **HIGH PERFORMANCE E²CMOS® TECHNOLOGY**
  - **— 20 ns Maximum Propagation Delay**
  - **— Fmax = 41.6 MHz**
  - **— 90 mA MAX Icc**
- **E² CELL TECHNOLOGY**
  - **— 100% Tested/Guaranteed 100% Yields**
  - **— 20 Year Data Retention**
- **EIGHT OUTPUT LOGIC MACROCELLS**
  - **— Maximum Flexibility for Complex Logic Designs**
  - **— Programmable Output Polarity**
  - **— Also Emulates 20-pin PAL® Devices with Full Function/Fuse Map/Parametric Compatibility**
- **PRELOAD AND POWER-ON RESET OF REGISTERS**
  - **— 100% Functional Testability**
- **APPLICATIONS INCLUDE:**
  - **— Reconfigurable Interfaces and Decoders**
  - **— "Soft" Hardware (Generic Systems)**
  - **— Copy Protection and Security Schemes**
  - **— Reconfiguring Systems for Testing**
- **ELECTRONIC SIGNATURE FOR IDENTIFICATION**

FUNCTIONAL BLOCK DIAGRAM



DESCRIPTION

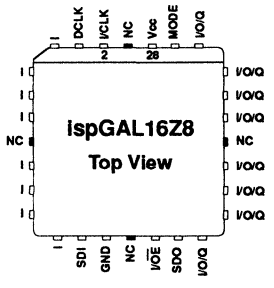
The Lattice ispGAL® 16Z8 is a revolutionary programmable logic device featuring 5-volt only in-system programmability and in-system diagnostic capabilities. This is made possible by on-chip circuitry which generates and shapes the necessary high voltage programming signals. Using Lattice's proprietary UltraMOS® technology, this device provides true bipolar performance at significantly reduced power levels.

The 24-pin ispGAL16Z8 is architecturally and parametrically identical to the 20-pin GAL16V8, but includes 4 extra pins to control in-system programming. These pins are not associated with normal logic functions and are used only during programming and diagnostic operations. This 4-pin interface allows an unlimited number of devices to be cascaded to form a serial programming and diagnostics loop.

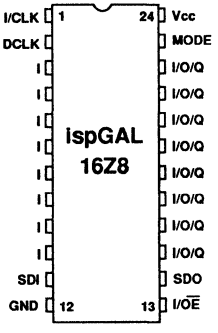
Unique test circuitry and reprogrammable cells allow complete AC, DC, and functional testing during manufacture. Therefore, LATTICE is able to guarantee 100% field programmability and functionality of all GAL® products.

PIN CONFIGURATION

PLCC



DIP



LATTICE SEMICONDUCTOR CORP., 5555 N.E. Moore Ct., Hillsboro, Oregon 97124, U.S.A.
Tel. (503) 681-0118 or 1-800-FASTGAL; FAX (503) 681-3037

December 1991

ABSOLUTE MAXIMUM RATINGS[1]

Supply voltage V_{CC} ..... -0.5 to +7V
Input voltage applied ..... -2.5 to V_{CC} +1.0V
Off-state output voltage applied ..... -2.5 to V_{CC} +1.0V
Storage Temperature ..... -65 to 150°C
Ambient Temperature with
Power Applied ..... -55 to 125°C

1. Stresses above those listed under the "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress only ratings and functional operation of the device at these or at any other conditions above those indicated in the operational sections of this specification is not implied (while programming, follow the programming specifications).

RECOMMENDED OPERATING COND.

Ambient Temperature (T_A) ..... 0 to +75°C
Supply voltage (V_{CC})
with Respect to Ground ..... +4.75 to +5.25V

DC ELECTRICAL CHARACTERISTICS

Over Recommended Operating Conditions (Unless Otherwise Specified)

SYMBOL	PARAMETER	CONDITION	MIN.	TYP.[2]	MAX.	UNITS
V_{IL}	Input Low Voltage		Vss – 0.5	—	0.8	V
V_{IH}	Input High Voltage		2.0	—	Vcc+1	V
I_{IL}	Input or I/O Low Leakage Current	$0V \le V_{IN} \le V_{IL}$ (MAX.)	—	—	–10	µA
I_{IH}	Input or I/O High Leakage Current	$V_{IH} \le V_{IN} \le V_{CC}$	—	—	10	µA
V_{OL}	Output Low Voltage	I_{OL} = MAX. Vin = V_{IL} or V_{IH}	—	—	0.5	V
V_{OH}	Output High Voltage	I_{OH} = MAX. Vin = V_{IL} or V_{IH}	2.4	—	—	V
I_{OL}	Low Level Output Current		—	—	24	mA
I_{OH}	High Level Output Current		—	—	–3.2	mA
I_{OS}[1]	Output Short Circuit Current	V_{CC} = 5V V_{OUT} = 0.5V T_A = 25°C	–30	—	–150	mA
ICC	Operating Power Supply Current	VIL = 0.5V VIH = 3.0V ftoggle = 15Mhz Outputs Open	—	75	90	mA

1) One output at a time for a maximum duration of one second. Vout = 0.5V was selected to avoid test problems caused by tester ground degradation. Guaranteed but not 100% tested.
2) Typical values are at Vcc = 5V and T_A = 25 °C

CAPACITANCE (T_A = 25°C, f = 1.0 MHz)

SYMBOL	PARAMETER	MAXIMUM*	UNITS	TEST CONDITIONS
C_I	Input Capacitance	8	pF	V_{CC} = 5.0V, V_I = 2.0V
$C_{I/O}$	I/O Capacitance	10	pF	V_{CC} = 5.0V, $V_{I/O}$ = 2.0V

*Guaranteed but not 100% tested.

AC SWITCHING CHARACTERISTICS

Over Recommended Operating Conditions

PARAMETER	TEST COND.[1]	DESCRIPTION	-20		-25		UNITS
			MIN.	MAX.	MIN.	MAX.	
tpd	1	Input or I/O to Combinational Output	3	20	3	25	ns
tco	1	Clock to Output Delay	2	15	2	15	ns
tsu	—	Setup Time, Input or Feedback before Clock↑	15	—	20	—	ns
th	—	Hold Time, Input or Feedback after Clock↑	0	—	0	—	ns
fmax[2]	1	Maximum Clock Frequency with External Feedback, 1/(tsu + tco)	33.3	—	28.5	—	MHz
	1	Maximum Clock Frequency with No Feedback	41.6	—	33.3	—	MHz
twh[3]	—	Clock Pulse Duration, High	12	—	15	—	ns
twl[3]	—	Clock Pulse Duration, Low	12	—	15	—	ns
ten	2	Input or I/O to Output Enabled	—	20	—	25	ns
	2	OE↓ to Output Enabled	—	18	—	20	ns
tdis	3	Input or I/O to Output Disabled	—	20	—	25	ns
	3	OE↑ to Output Disabled	—	18	—	20	ns

1) Refer to **Switching Test Conditions** section.
2) Refer to **fmax Description** section.
3) Clock pulses of widths less than the specification may be detected as valid clock signals.

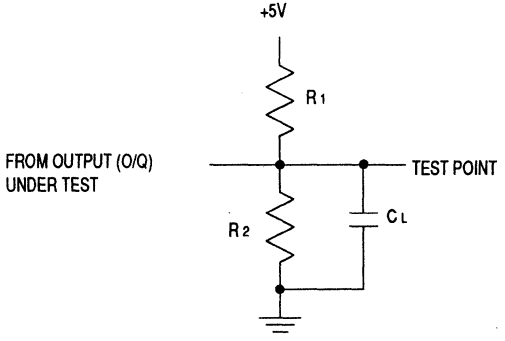
SWITCHING TEST CONDITIONS

Input Pulse Levels	GND to 3.0V
Input Rise and Fall Times	3ns 10% – 90%
Input Timing Reference Levels	1.5V
Output Timing Reference Levels	1.5V
Output Load	See Figure

3-state levels are measured 0.5V from steady-state active level.

Output Load Conditions (see figure)

Test Condition		R1	R2	CL
1		200Ω	390Ω	50pF
2	Active High	∞	390Ω	50pF
	Active Low	200Ω	390Ω	50pF
3	Active High	∞	390Ω	5pF
	Active Low	200Ω	390Ω	5pF



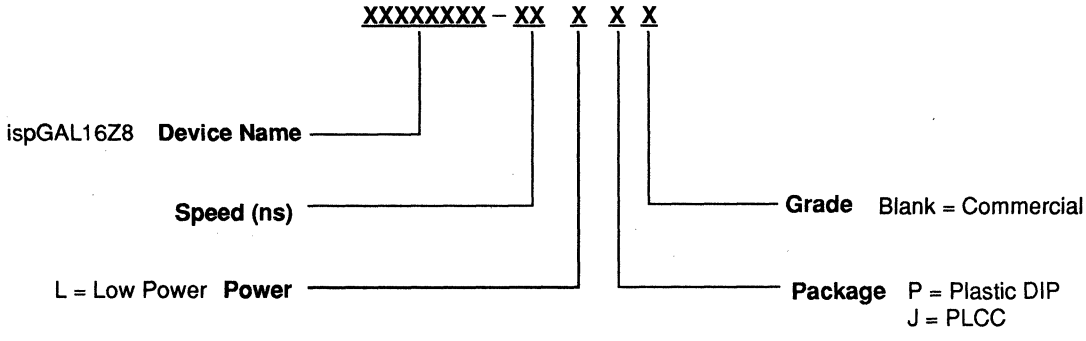
C L INCLUDES JIG AND PROBE TOTAL CAPACITANCE

ispGAL16Z8 ORDERING INFORMATION

Commercial Grade Specifications

Tpd (ns)	Tsu (ns)	Tco (ns)	Icc (mA)	Ordering #	Package
20	15	15	90	ispGAL16Z8-20LP	24-Pin Plastic DIP
			90	ispGAL16Z8-20LJ	28-Lead PLCC
25	20	15	90	ispGAL16Z8-25LP	24-Pin Plastic DIP
			90	ispGAL16Z8-25LJ	28-Lead PLCC

PART NUMBER DESCRIPTION



Military GAL Device Summary

Lattice offers the most comprehensive line of military E²CMOS Programmable Logic Devices. Lattice recognizes the trend in military device procurement towards using SMD compliant devices and encourages customers to use the SMD number, where it exists, when ordering parts. Listed below are Lattice's military qualified devices and their corresponding SMD numbers. Please contact your local Lattice representative for the latest product listing.

MILITARY PRODUCTS SELECTOR GUIDE

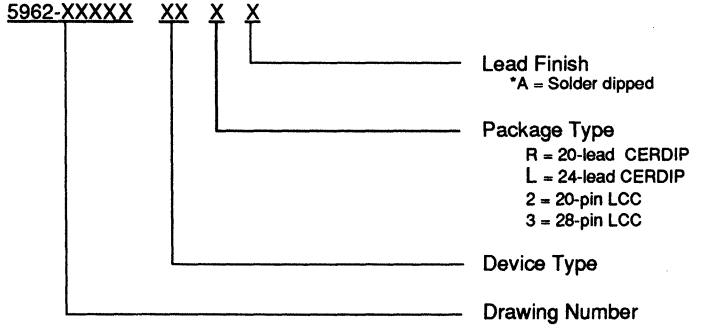
DEVICE TYPE	Tpd (ns)	Icc (mA)	PACKAGE	LATTICE PART #	SMD #
GAL16V8	10	130	20-Pin CERDIP	GAL16V8B-10LD/883C	5962-8983904RA
		130	20-Pin LCC	GAL16V8B-10LR/883C	5962-89839042A
	15	130	20-Pin CERDIP	GAL16V8A-15LD/883C	5962-8983903RA
		130	20-Pin LCC	GAL16V8A-15LR/883C	5962-89839032A
	20	65	20-Pin CERDIP	GAL16V8A-20QD/883C	5962-8983906RA
		65	20-Pin LCC	GAL16V8A-20QR/883C	5962-89839062A
		130	20-Pin CERDIP	GAL16V8A-20LD/883C	5962-8983902RA
		130	20-Pin LCC	GAL16V8A-20LR/883C	5962-89839022A
	25	65	20-Pin CERDIP	GAL16V8A-25QD/883C	5962-8983905RA
		65	20-Pin LCC	GAL16V8A-25QR/883C	5962-89839052A
	30	130	20-Pin CERDIP	GAL16V8A-30LD/883C	5962-8983901RA
		130	20-Pin LCC	GAL16V8A-30LR/883C	5962-89839012A
GAL20V8	15	130	24-Pin CERDIP	GAL20V8A-15LD/883C	5962-8984003LA
		130	28-Pin LCC	GAL20V8A-15LR/883C	5962-89840033A
	20	65	24-Pin CERDIP	GAL20V8A-20QD/883C	Contact Factory
		65	28-Pin LCC	GAL20V8A-20QR/883C	Contact Factory
		130	24-Pin CERDIP	GAL20V8A-20LD/883C	5962-8984002LA
		130	28-Pin LCC	GAL20V8A-20LR/883C	5962-89840023A
	25	65	24-Pin CERDIP	GAL20V8A-25QD/883C	Contact Factory
		65	28-Pin LCC	GAL20V8A-25QR/883C	Contact Factory
	30	130	24-Pin CERDIP	GAL20V8A-30LD/883C	5962-8984001LA
		130	28-Pin LCC	GAL20V8A-30LR/883C	5962-89840013A
GAL22V10	15	150	24-Pin CERDIP	GAL22V10-15LD/883C	5962-8984103LA
		150	28-Pin LCC	GAL22V10-15LR/883C	5962-89841033A
	20	150	24-Pin CERDIP	GAL22V10-20LD/883C	5962-8984102LA
		150	28-Pin LCC	GAL22V10-20LR/883C	5962-89841023A
	25	150	24-Pin CERDIP	GAL22V10-25LD/883C	5962-8984104LA
		150	28-Pin LCC	GAL22V10-25LR/883C	5962-89841043A
	30	150	24-Pin CERDIP	GAL22V10-30LD/883C	5962-8984101LA
		150	28-Pin LCC	GAL22V10-30LR/883C	5962-89841013A
GAL26CV12	20	160	28-Pin CERDIP	GAL26CV12-20LD/883C	Contact Factory
		160	28-Pin LCC	GAL26CV12-20LR/883C	Contact Factory
	25	160	28-Pin CERDIP	GAL26CV12-25LD/883C	Contact Factory
		160	28-Pin LCC	GAL26CV12-25LR/883C	Contact Factory
GAL20RA10	20	120	24-Pin CERDIP	GAL20RA10-20LD/883C	Contact Factory
		120	28-Pin LCC	GAL20RA10-20LR/883C	Contact Factory
	25	120	24-Pin CERDIP	GAL20RA10-25LD/883C	Contact Factory
		120	28-Pin LCC	GAL20RA10-25LR/883C	Contact Factory

DESC STANDARD MILITARY DRAWING LISTING

SMD #	LATTICE PART #
5962-89839012A	GAL16V8A-30LR/883C
5962-8983901RA	GAL16V8A-30LD/883C
5962-89839022A	GAL16V8A-20LR/883C
5962-8983902RA	GAL16V8A-20LD/883C
5962-89839032A	GAL16V8A-15LR/883C
5962-8983903RA	GAL16V8A-15LD/883C
5962-89839042A	GAL16V8B-10LR/883C
5962-8983904RA	GAL16V8B-10LD/883C
5962-89839052A	GAL16V8A-25QR/883C
5962-8983905RA	GAL16V8A-25QD/883C
5962-89839062A	GAL16V8A-20QR/883C
5962-8983906RA	GAL16V8A-20QD/883C
5962-89840013A	GAL20V8A-30LR/883C

SMD #	LATTICE PART #
5962-8984001LA	GAL20V8A-30LD/883C
5962-89840023A	GAL20V8A-20LR/883C
5962-8984002LA	GAL20V8A-20LD/883C
5962-89840033A	GAL20V8A-15LR/883C
5962-8984003LA	GAL20V8A-15LD/883C
5962-89841013A	GAL22V10-30LR/883C
5962-8984101LA	GAL22V10-30LD/883C
5962-89841023A	GAL22V10-20LR/883C
5962-8984102LA	GAL22V10-20LD/883C
5962-89841033A	GAL22V10-15LR/883C
5962-8984103LA	GAL22V10-15LD/883C
5962-89841043A	GAL22V10-25LR/883C
5962-8984104LA	GAL22V10-25LD/883C

STANDARD MILITARY DRAWING NUMBER DESCRIPTION



* no other lead finish currently available.

Section 1: Introduction to pLSI™ and ispLSI™

Section 2: pLSI and ispLSI Data Sheet

Section 3: GAL® Data Specifications

Section 4: pLSI and ispLSI Architecture

Introduction 4-1

General Block Diagram 4-1

Generic Logic Block 4-3

I/O Cell 4-6

Output Routing Pool 4-8

Global Routing Pool 4-10

Megablock 4-10

Clock Distribution Network 4-12

Timing Model 4-14

ispLSI Programming Information 4-23

Section 5: pLSI and ispLSI Advantages

Section 6: pLSI and ispLSI Software Development Tools

Section 7: pLSI and ispLSI Design Optimization

Section 8: pLSI and ispLSI Application Notes

Section 9: pLSI and ispLSI Article Reprints

Section 10: pLSI and ispLSI Quality, Reliability & Programmer Support

Section 11: Sales Information

pLSI and ispLSI Architecture

Introduction

This section provides a detailed description of the five major architectural building blocks which make up the pLSI and ispLSI device families. In addition, the timing model for these families is covered, as well as information on programming ispLSI devices.

The pLSI and ispLSI device families consist of 8 devices. Referring to table 4-1, it can be seen that there are four device sizes ranging from 16 to 48 GLBs. The ispLSI devices differ from the pLSI devices only in the area of in-system programming. The ispLSI family supports the Lattice in-system programming interface for designs which need reconfigurability in the system.

General Block Diagram

The Lattice pLSI and ispLSI devices are High Density Programmable Logic Devices which contain programmable logic, registers, I/O pins, multiple clocks, a Global Routing Pool (GRP) and an Output Routing Pool (ORP). The GRP allows complete interconnectivity between all of these elements (see figure 4-1).

The basic logic unit is the Generic Logic Block (GLB). The GLBs are labeled A0 to D7 (see figure 4-1). A total of 32 GLBs are contained in the pLSI 1032 and ispLSI 1032 devices. Each GLB has 18 inputs, a programmable AND/OR/XOR array, and four outputs. The outputs are configured as either combinatorial or registered. Inputs to GLBs come from the GRP and dedicated input pins. All GLB outputs are available to the GRP so they can be connected to the inputs of other GLBs.

The pLSI and ispLSI devices also have I/O Cells, each of which is directly connected to an I/O pin. Each I/O cell can be individually programmed to be a combinatorial input, a registered input, a latched input, a direct output, a 3-state output or a bidirectional I/O pin. The signal levels are TTL compatible levels. The Output drivers can source 4 mA and sink 8 mA. The I/O Cells are grouped in sets of 16. Each set of 16 outputs is routed through an ORP for pin flexibility and share a common Output Enable signal.

Each device also has Dedicated Input Pins. The Dedicated Input Pins connect to two of the 18 inputs to the GLBs.

Eight GLBs, 16 I/O Cells and one ORP are connected together to make a logic Megablock. The Megablock is defined by the resources that it shares. The outputs of the eight GLBs are connected to a set of 16 universal I/O cells by the ORP. The pLSI 1032 device contains four of these Megablocks (see figure 4-2).

The GRP inputs come from the outputs of the GLBs and the external inputs from the bidirectional I/O cells. Delays through the GRP are equalized to minimize timing skew and logic glitching.

Clocks for the pLSI and ispLSI devices are selected using the Clock Distribution Network. Four dedicated clock pins are brought into the Clock Distribution Network and five outputs are provided to route clocks to the GLBs and I/O cells. The Clock Distribution Network is also closely associated with a special clock GLB. The logic of this GLB can be used to create an internal clock from any combination of signals within the device.

Table 4-1. pLSI and ispLSI Device Families

	pLSI 1016 ispLSI 1016	pLSI 1024 ispLSI 1024	pLSI 1032 ispLSI 1032	pLSI 1048 ispLSI 1048
GLBs	16	24	32	48
I/Os + I	32 + 4	48 + 6	64 + 8	96 + 10
Registers	96	144	192	288
GRPs	1	1	1	1
ORPs	2	3	4	6
Global Clocks	5	5	5	5

Figure 4-1. Functional Block Diagram pLSI and ispLSI 1032

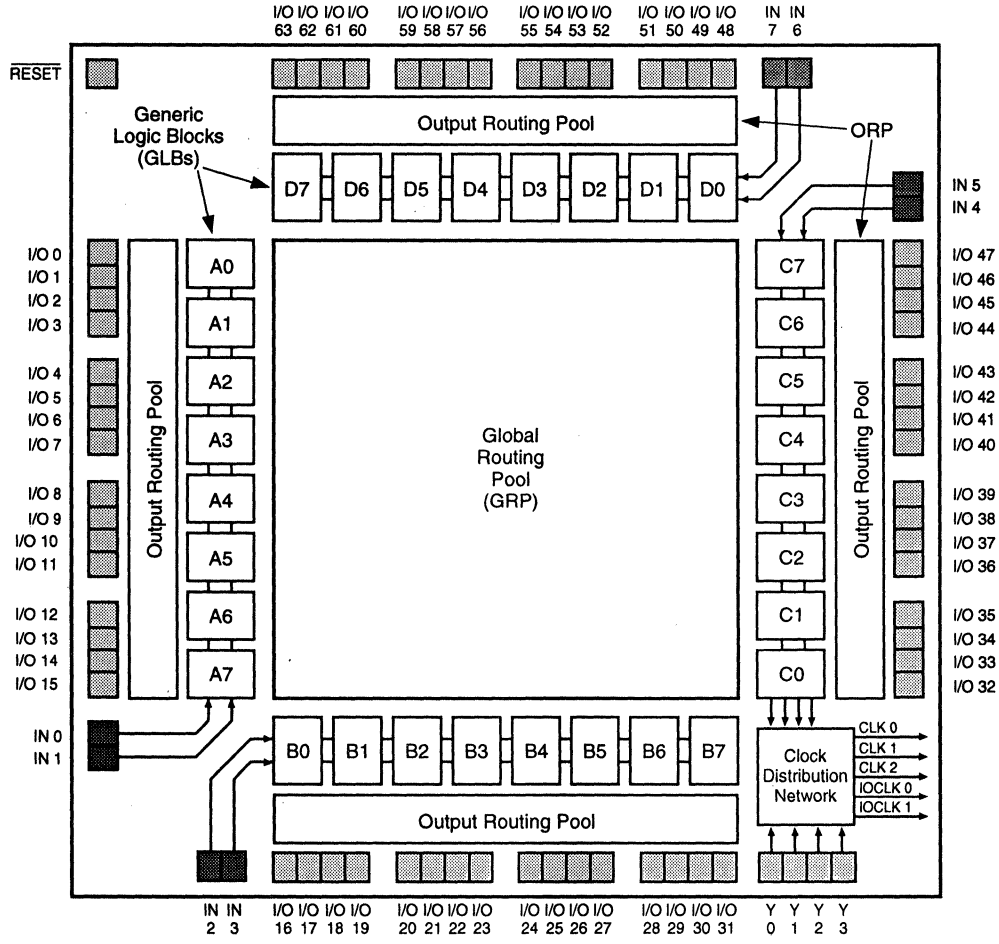
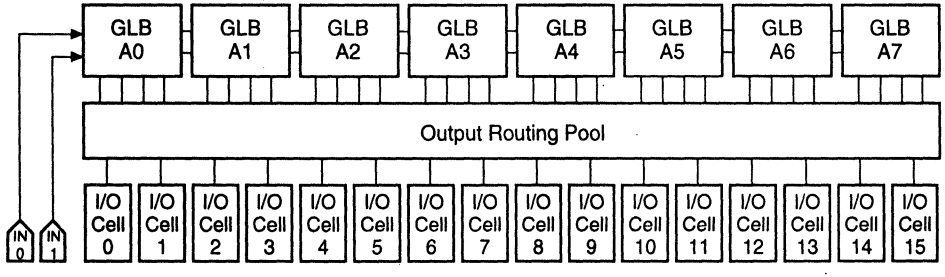


Figure 4-2. Megablock Diagram



Note: The inputs from the I/O cell are not shown in this figure, they go directly to the GRP.

Generic Logic Block

The GLB is the standard logic building block of the Lattice pLSI and ispLSI device family. The GLB has 18 inputs, four outputs, four D-type registers, and the logic necessary to implement 90% of all 4-bit logic functions (see figure 4-3).

The GLB internal logic is divided into four separate sections:

- AND Array
- Product Term Sharing Array (PTSA)
- Output Logic Macrocells (OLMC)
- Control Function

AND Array

The AND Array consists of 18 GLB inputs which provide both a true and a complement for each signal. These signals drive an E²CMOS array of programmable cells.

The cell array interconnects the signals with the 20 product terms (PTs) in the GLB. The PTs produce the logical sum of the 18 GLB inputs.

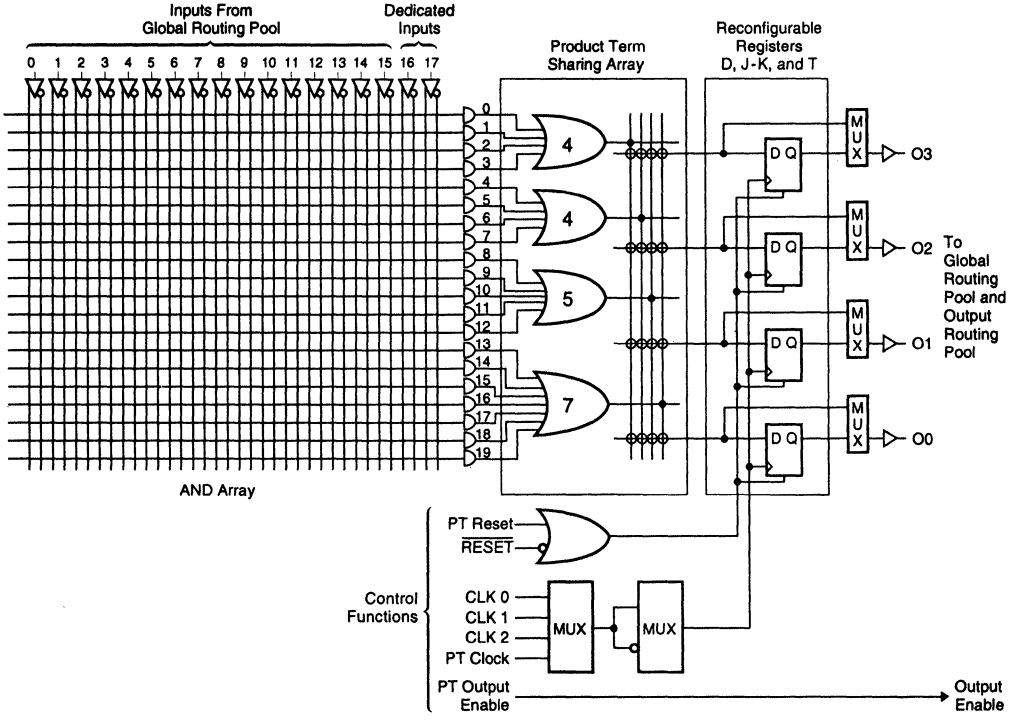
Sixteen of the GLB inputs come from the Global Routing Pool (GRP), and are either feedback signals from the GLBs or inputs from the external I/O Cells. The two remaining inputs come directly from the two dedicated input pins associated with the Megablock.

Product Term Sharing Array

The Product Term Sharing Array (PTSA) makes the 20 PTs available to the GLB outputs as needed by the logic design. The PT sum terms are grouped into four OR gates containing 4, 4, 5 and 7 inputs respectively (see figure 4-3).

The output of these gates are connected to the 4 GLB outputs through the PTSA. If the design requires more PTs than the basic 4, 5, or 7, the PTSA combines them to provide up to 20 PTs per output of the GLB.

Figure 4-3. Simplified GLB Logic Diagram



4 Product Term Bypass

If the main concern is speed, the PTSA can be bypassed with a four-input OR gate which provides up to 4 PTs to the output. This bypass is called the 4-PT Bypass and reduces the delay associated with the PTSA and the XOR gate (see figure 4-4).

This feature simplifies designing counters, comparators, and ALU type functions. The OLMCs can be bypassed after the XOR gate, if a combinatorial output is needed. Each GLB output returns to the GRP and to the I/O cells via the ORP. The XOR gates are not available when the four-product term bypass is used.

Output Logic Macrocells

The GLB Output Logic Macro Cells (OLMCs) consist of four D-type flip-flops with XOR gates on their inputs. The XOR gates are used either as logic elements or to reconfigure the D-type flip-flops to emulate J-K or T-type flip-flops (see figure 4-5).

The PTSA is flexible enough to allow these features to be used in virtually any combination that is required. In the GLB shown in figure 4-6, O3 is configured using the XOR Gate and O2 is configured using the 4 Product Term Bypass. O1 uses one of the inputs from the five-product term OR gate. O0 combines the remaining four product terms with the product terms from the seven-product term OR gate for a total of eleven (7+4).

Figure 4-4. GLB Showing a 4-PT Bypass

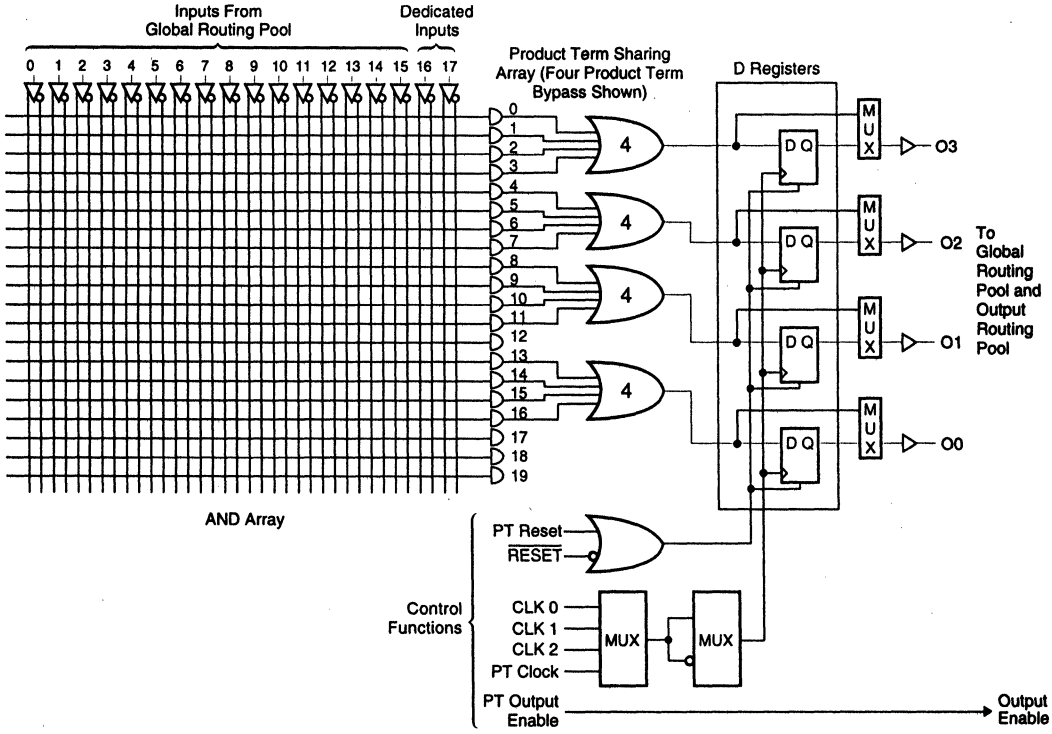


Figure 4-5. GLB Showing XORs

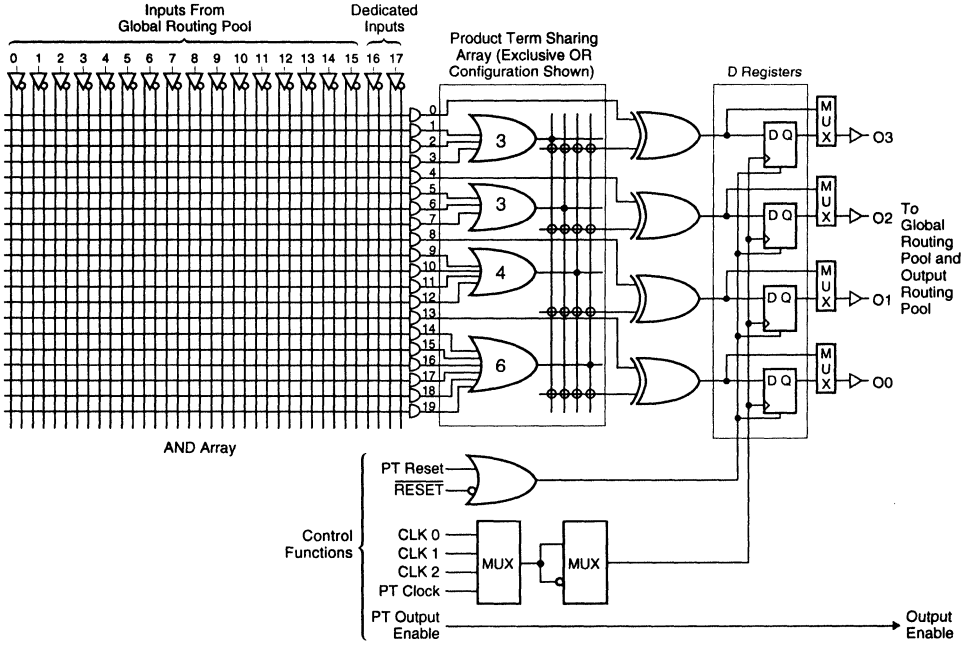
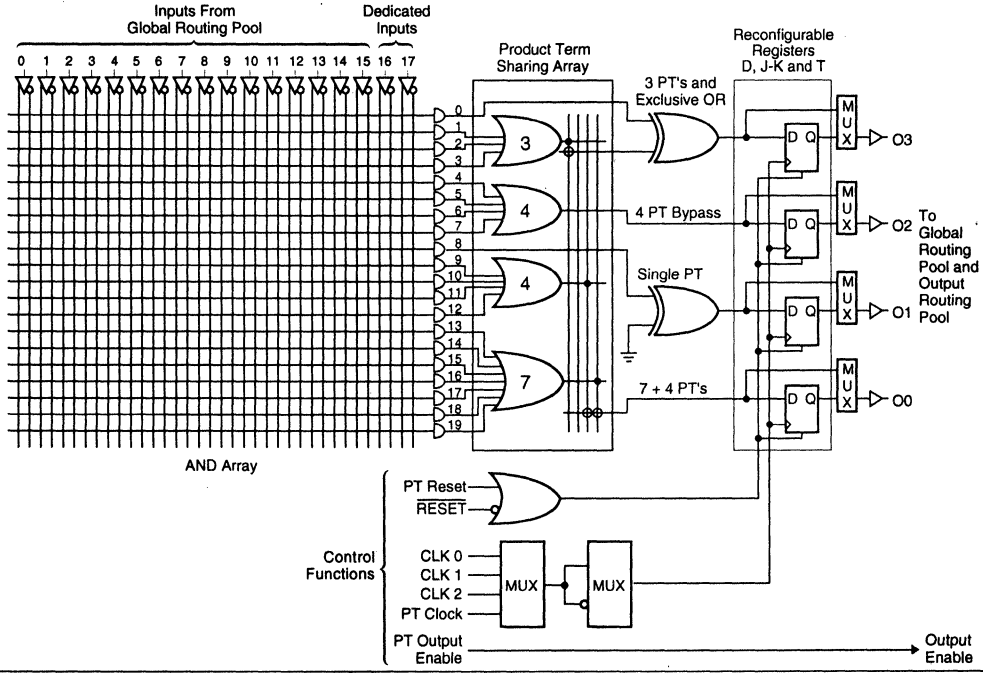


Figure 4-6. GLB Showing Various Configurations

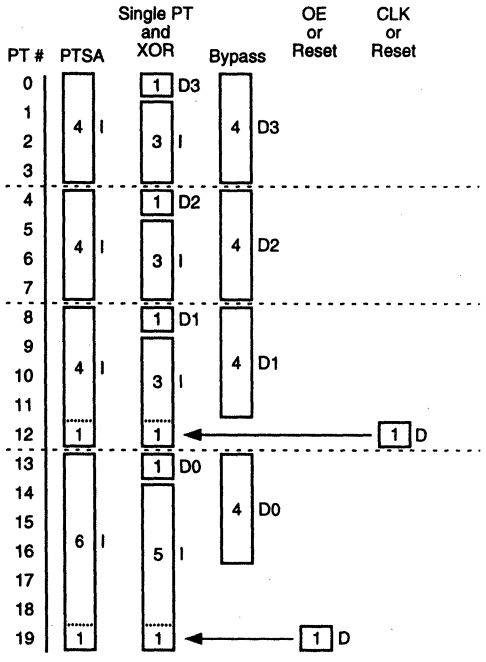


Control Functions

Signals that control the operation of the GLBs are developed in the Control area. The clock for the registers comes from one of three clock lines developed in the Clock Distribution Network (Synchronous) or from PT 12 within the GLB (Asynchronous). The Reset signal for the GLB comes from the Global Reset pin or from PT 12 or PT 19 within the GLB. The Output Enable for the I/O cells associated with the GLB comes from PT 19 within the GLB. If the design needs to use a product term for a control function, that product term becomes unavailable for use as a logic term. Refer to table 4-2 to determine which logic functions are affected.

Table 4-2 highlights how the product terms are utilized in the various modes. As an example, Product Term 12 is used as a single input to the five-input OR gate in the standard configuration. This OR gate can also be routed to the four GLB outputs, if it is not used in the four-product term bypass mode. When GLB output O1 is used in the XOR mode, PT 12 becomes one of the inputs to the four-input OR Gate.

Table 4-2. Product Term Grouping



I = Indirect Thru PTSA
D = Dedicated To Specific OLMC

If PT 12 is not used in the logic, it is available as either the PT Clock signal or the PT Reset signal.

I/O Cell

The pLSI and ispLSI devices contain several general purpose I/O pins. Every I/O pin has an I/O cell associated with it. The I/O cells can be configured for different functions such as input registers and input latches.

Table 4-3 illustrates the available number of I/O pins for different pLSI and ispLSI devices. I/O pins connect the signals within the pLSI devices to the outside (inputs and outputs to the device).

Most competitors' devices do not have enough I/O pins for the logic inside the device and therefore, cannot properly interface to the outside world. In comparison, pLSI and ispLSI devices have an I/O ratio of one pin for every two GLB output signals.

Table 4-3. Available I/O Pins for pLSI and ispLSI Devices

Device	Number of I/O Pins
pLSI 1016, ispLSI 1016	32
pLSI 1024, ispLSI 1024	48
pLSI 1032, ispLSI 1032	64
pLSI 1048, ispLSI 1048	96

I/O Cell Configurations

Figure 4-7 illustrates the I/O cell architecture. I/O cells can be configured as 31 different functions. Table 4-4 shows some of these possible combinations. Figure 4-8 shows the schematics for some of the configurations.

Table 4-4. I/O Cell Configurations

Type	Grouping
Inputs	Registered Latched Combinatorial
Output	Combinatorial Active High Combinatorial Active Low
Bidirectional	Registered Input Latched Input Combinatorial Input

I/O Cell Architecture Description

In figure 4-7, the output enable signal (OE) comes from the output enable scheme. The OE's polarity is controlled by a multiplexer. Two input signals come from the ORP. One of these signals comes from the ORP bypass, the other from the standard ORP. The bypass signal is selected for faster propagation delay (t_{pd}) and clock to output (t_{co}) performance.

Every I/O cell has an input register that can be configured either as a flip-flop or a Transparent Latch. The control signal, labeled R/L on the schematic, configures the register. The external input signal feeds into the input of this register. The output feeds directly into the GRP. For combinatorial inputs, the register is bypassed using a multiplexer.

For bidirectional signals, any of the three input configurations, i.e. combinatorial, registered or latched input, can be combined with the combinatorial output under the control of the output enable signal.

There are also two clocks in each I/O cell that are positive or negative (edge or level) triggered. These clocks are labeled as IOCLK 0 and IOCLK 1. The clocks are part of a global clock network and are available to all the I/O cells in the device. The Global Reset feeds into every I/O cell. It clears the registers on power up. An active low Reset signal resets the register within the I/O cell to the low state.

The active pullup resistor in the I/O cell is automatically enabled when the pin is not used. This prevents the pin from floating and potentially inducing noise into the circuit or consuming additional power.

4

Figure 4-7. I/O Cell Architecture

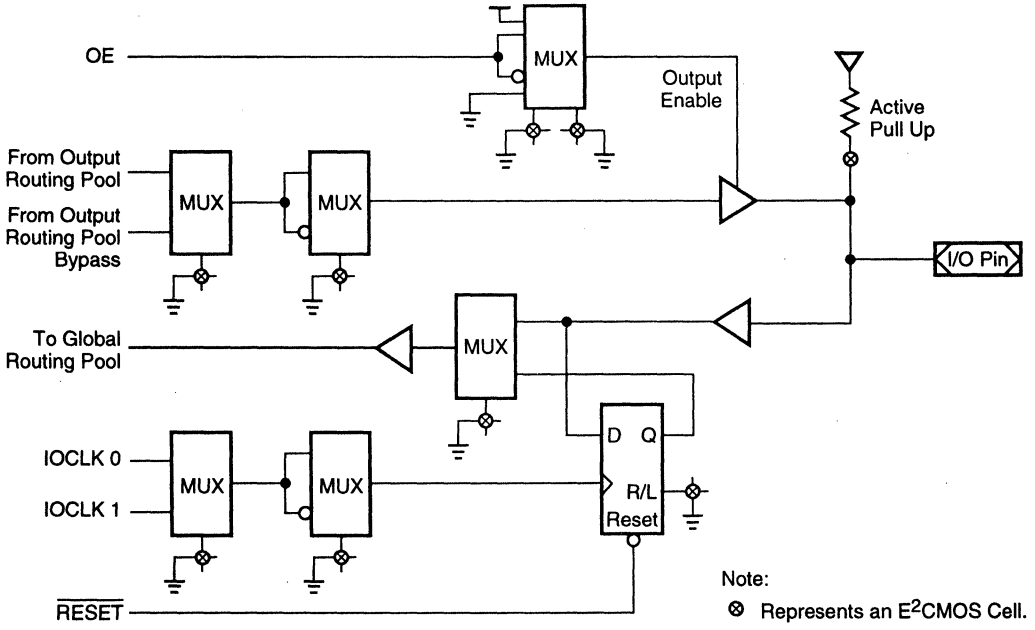
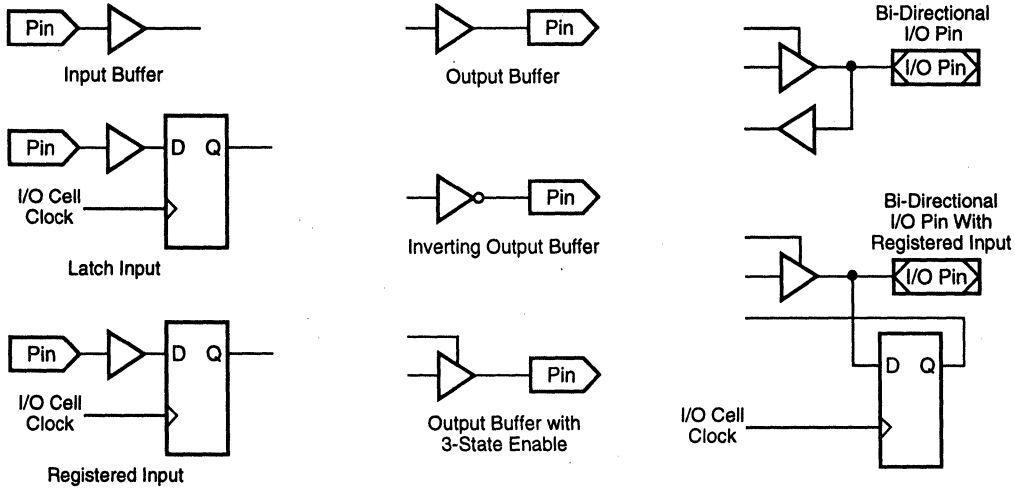


Figure 4-8. Schematic of Possible Configurations



Output Routing Pool

The Output Routing Pool (ORP) performs the following functions:

- Routes Signals from GLBs to I/Os
- Provides Flexibility of Pin Assignment
- Provides Bypass Paths for Fast Signals
- Provides Flexibility for Fixing Signals to Meet the Design Constraints
- Helps Optimize Routing and Enhances Logic Utilization
- Provides Easy Bussing in Groups of 2, 4, 8, or 16 Bits

Eight GLBs and 16 I/Os share an ORP (see figure 4-9). The ORP is one of the main resources of a Megablock. Each Megablock in the pLSI or ispLSI device consists of an ORP, 16 I/O cells, 8 GLBs, 2 dedicated inputs and a common OE. The ORP is used to interconnect the GLB outputs to the I/O cells. The ORP enhances the flexibility of the pin assignment by routing signals to several different pins. It also provides ORP bypass paths for critical signals.

Figure 4-10 illustrates the flexibility of the pin assignment by routing every GLB output within a Megablock to four different locations.

Figure 4-9. Shared ORP

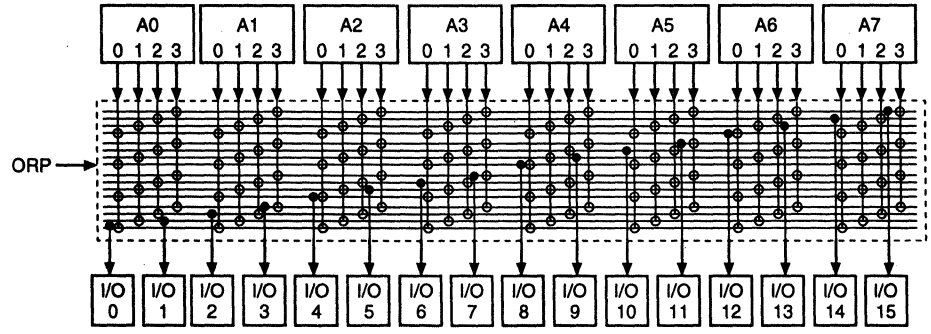
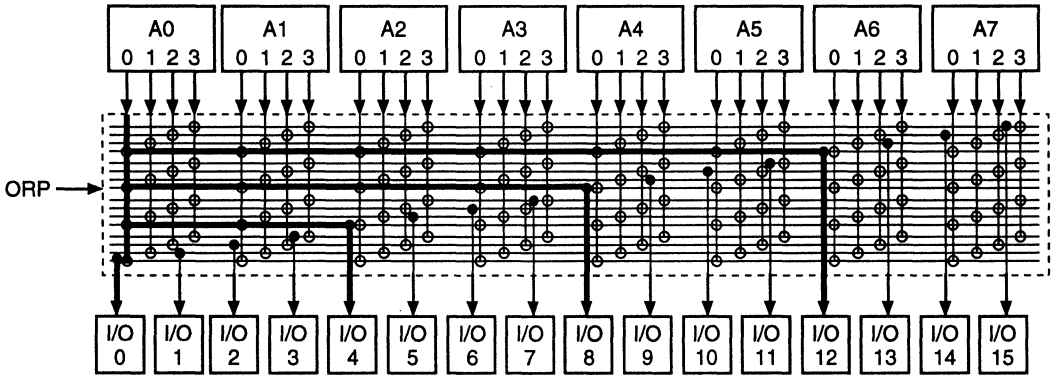


Figure 4-10. ORP Flexibility



Output signal zero is routed from GLB A0 to I/O cell zero, four, eight, and twelve. Similarly other signals can be routed to four different locations by going through the ORP interconnects.

An ORP bypass is primarily used for fast signals. Additionally the ORP bypass signals should be used for fast signals only, to avoid creating routing bottlenecks.

Figure 4-11. ORP With Bypass

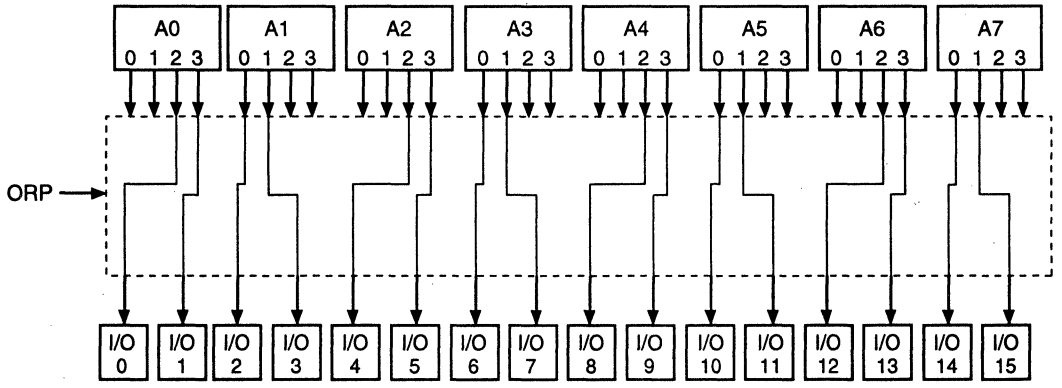
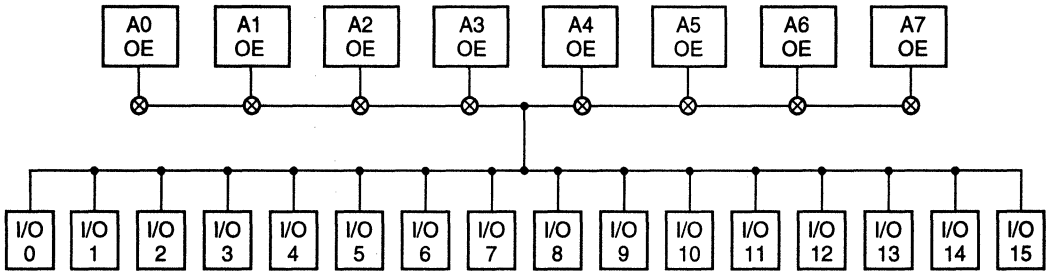


Figure 4-11 also illustrates the configuration of two bypass signals per GLB for a total of 16 signals in a group of 8 GLBs (Megablock). This feature accommodates bussing in groups of two bits instead of four which is the standard ORP configuration when a bypass is not used.

Output Enable Scheme

Figure 4-12 illustrates the Output Enable (OE) used on pLSI and ispLSI devices. Any of the eight GLBs can generate the OE which can be used by any of the 16 I/O cells within the megablock.

Figure 4-12. Output Enable Scheme



Global Routing Pool

The Global Routing Pool (GRP) is the primary routing resource and performs the following functions:

- Provides Complete Interconnectivity for Highest Logic Utilization
- Provides Predictable Delays
- Simplifies Place and Route and Offers Quick Turn-around of Designs
- Connects All I/O Pin Inputs to the GLB Inputs
- Connects All GLB Outputs to Other GLB Inputs

Lattice provides fully automatic Place and Route software which eliminates time-consuming manual editing.

Routing Structure

The pLSI and ispLSI routing structure is a centralized interconnect array which consists of :

- GLB Inputs
- GLB Outputs
- Input From All I/O Cells

The functionality of GRP is two fold:

- Connectivity of All GLB Outputs to the Inputs of Other GLBs
- Connectivity of All I/O Pin Inputs to the GLBs

The GRP Inputs are from the I/O cell Inputs and GLB Outputs. Using a matrix of E²CMOS cells, the GRP connects these signals to its outputs, which are the Inputs of all the GLBs.

Figure 4-13 illustrates this routing structure. The GRP outputs can be a function of all Global signals, whether originating from I/O pins, or feedback from GLBs. The advantage of this routing structure is its fixed and predictable routing delays. Hence, it is easy to predict the system speed prior to designing with pLSI or ispLSI devices.

Megablock

The pLSI and ispLSI families are structured into multiple Megablocks. Table 4-5 shows the available number of Megablocks for each pLSI and ispLSI device. Figure 4-14 shows Megablock resources.

Table 4-5. Megablocks for pLSI and ispLSI Devices

Device	Number of Megablocks
pLSI 1016, ispLSI 1016	2
pLSI 1024, ispLSI 1024	3
pLSI 1032, ispLSI 1032	4
pLSI 1048, ispLSI 1048	6

Each Megablock has the following resources:

- 8 GLBs
- 1 Output Enable Signal and an Output Enable Multiplexer
- 2 Dedicated Inputs
- An Output Routing Pool (ORP)
- 16 I/O Cells

Understanding how to configure these resources ensures successful routing of designs. Refer to the section on Design Optimization describing the implications of the Router.

Figure 4-13. Routing Structure

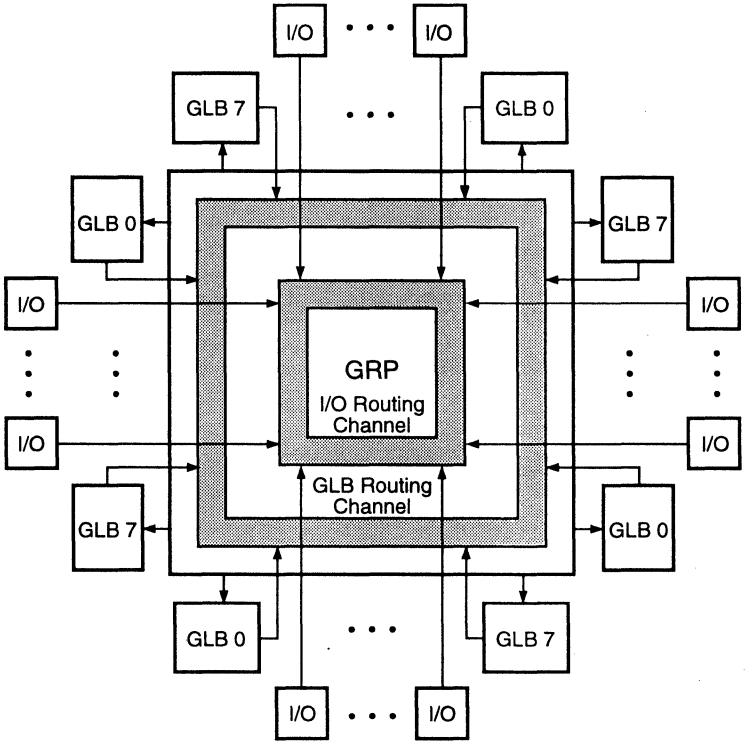
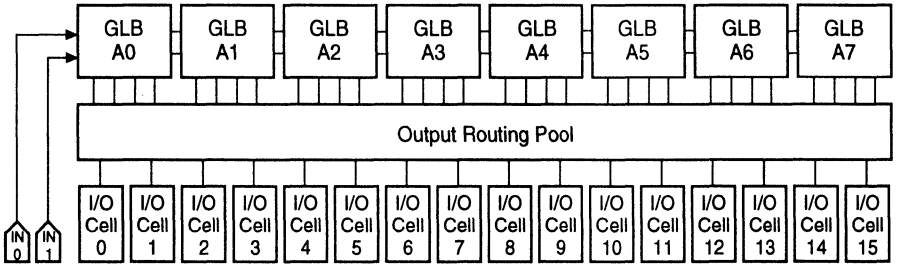


Figure 4-14. Megablock Resources

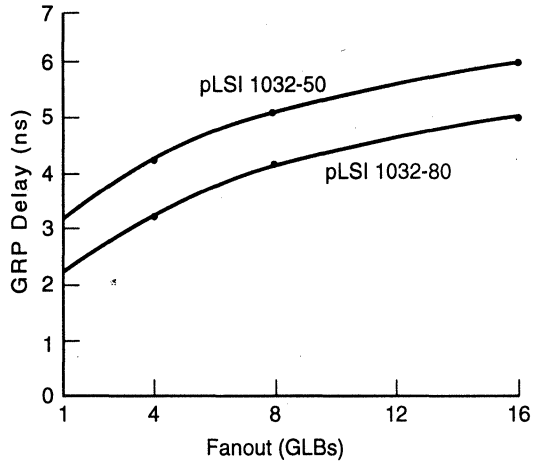


Note: The inputs from the I/O cell are not shown in this figure, they go directly to the GRP.

Delays vs Fanout

Fanout is defined as the number of GLB input signals being driven by the GRP. The interconnect delay increases according to the increase in fanout. To determine the performance of a design, calculate the worst case delay based upon the fanout of the signals (see figure 4-15).

Figure 4-15. Routing Channels



Clock Distribution Network

The pLSI and ispLSI devices offer a versatile clock scheme. This clock scheme is used to combine various subsystems of large designs into the same device. The devices offer three types of clocks:

- Synchronous Global Clocks
- Asynchronous Clocks Generated Within a GLB
- I/O Clocks Used With I/O Cells

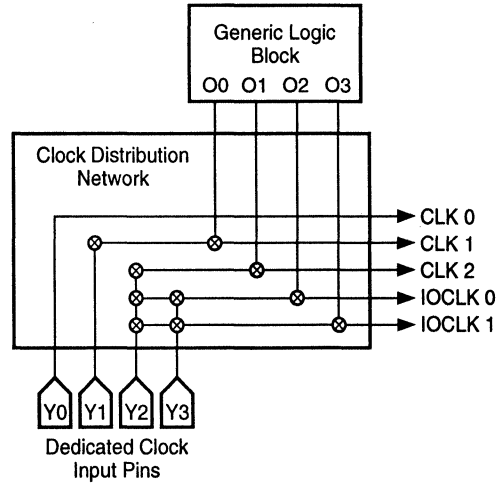
The following topics are discussed next:

- Global Clock Lines
- Dedicated Clock Pins
- Clock Generic Logic Block (GLB)
- Clock Polarity

Global Clock Lines

Three internal clock lines provide the global synchronous clock distribution to all the GLBs, and two clock lines are provided to the I/O cells (see figure 4-16). The global clock lines are available to all the GLBs and offer low skew. The I/O clock lines are selected by input cells which use the D-type flip-flops or latches. Like the GLB global clock lines, the global I/O clock lines are available to all I/O cells.

Figure 4-16. Clock Distribution Network of 1024, 1032 and 1048



Each device has a dedicated GLB associated with the clock network (see figure 4-16 and 4-17). This GLB can be used to place the output or final stage of logic used to generate global or synchronous GLB clocks. The Clock GLBs and their associated devices are listed in table 4-6.

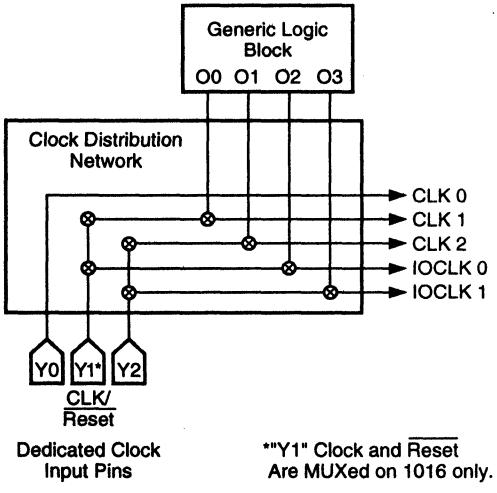
Table 4-6. Device and Clock GLBs

Device	Clock GLB
pLSI 1016, ispLSI 1016	B0
pLSI 1024, ispLSI 1024	B4
pLSI 1032, ispLSI 1032	C0
pLSI 1048, ispLSI 1048	D0

Dedicated Clock Pins

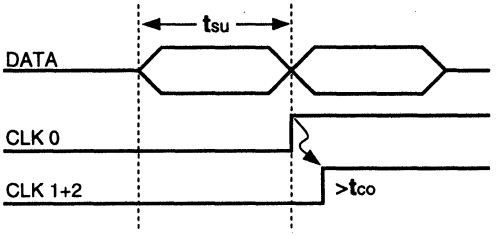
The internal GLB and I/O clock lines distribute signals from the clock GLB or from external pins. The pLSI and ispLSI 1024, 1032 and 1048 devices contain four external clock pins. The Y0 pin is hard wired to CLK 0 and improves performance. The pLSI and ispLSI 1016 devices have three external clock pins. The Y0 pin in each 1016 device is also hard wired to CLK 0 for enhanced performance. One of the two remaining pins, Y1, is connected to a Global Reset and a Global Clock Distribution Network (see figure 4-17).

Figure 4-17. Clock Distribution Network of pLSI 1016



In designs with more than one external clock, assign the most critical clock to the Y0 pin, since CLK 0 is hardwired and CLK 1 and CLK 2 are not hardwired (see figure 4-18).

Figure 4-18. Timing Diagram t_{co}



Clock GLB

One GLB is associated with the clock network. This GLB processes both standard logic, which is not associated with the global clock network, and the output portion of logic that creates clocks for use as synchronous GLB and I/O clocks.

With this feature, designers can use the GLB to clock multiple synchronous GLBs and I/Os from a clock that is generated from a combination of inputs both off chip or from other logic on chip. For example consider a system clock that operates at 66 MHz, and a small portion of the device must operate at the system clock speed while the rest of the device operates at half or some fraction of the input clock. This can be accomplished with the clock GLB.

Because there is only one GLB associated with the clock network, the final stage of all clock logic to be distributed through the global clock lines must be placed in the same GLB. The software automatically moves the logic to the correct clock GLB and connects the correct GLB to the clock network, no matter which device is being used. To use an internally generated global clock, define the clock signal names within one GLB and then use the same names in the logic GLBs.

Clock Polarity

A multiplexer is included in both GLBs and I/O cells. The multiplexer selects between the rising edge or the falling edge clock. The clock can be inverted or have its polarity changed. This is done for specific I/O cell clocks, GLB clocks, or global clocks when they reach their destinations.

Timing Model

The pLSI and ispLSI families of High Density Devices offer the fastest system speeds of any High Density Programmable Logic Device on the market today. To get the best performance out of pLSI and ispLSI parts, it is helpful to understand the various timing paths and parameters which are used in the part. By selecting fast logic paths for critical logic, and normal paths for slower logic, utilization and routability of the device can be increased, while still achieving maximum system performance.

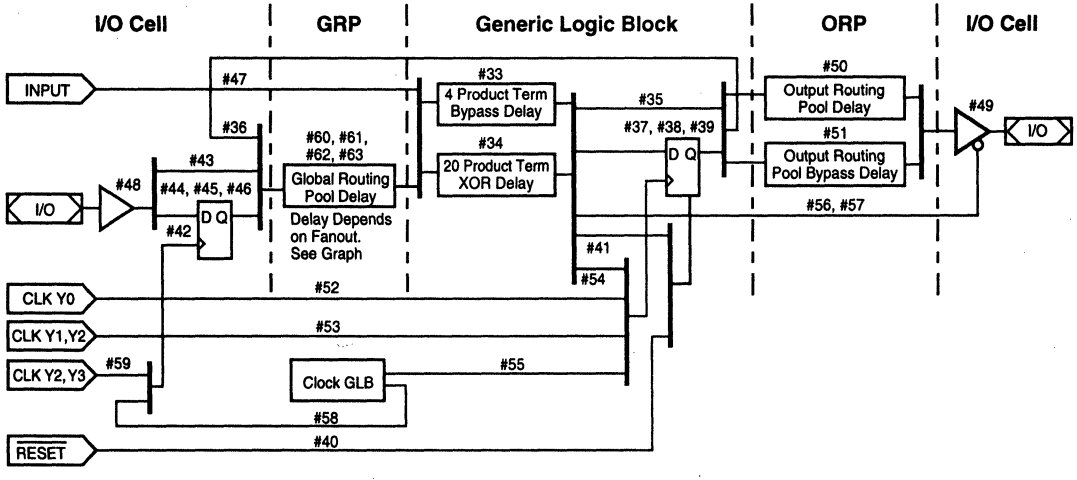
Timing parameters are divided into two categories: Internal Switching Characteristics and External Switching Characteristics.

Internal parameters are estimates of the times that signals will take to pass between the various logic blocks internal to the pLSI and ispLSI devices. Because it is not possible to test these individual internal signals, these values are specified in the data sheet for reference only.

External characteristics are parameters which can actually be measured by inserting a signal (or group of signals) on one pin and looking for the result on another pin. Each of these parameters are 100% tested on every device, and guaranteed to meet the data sheet specifications.

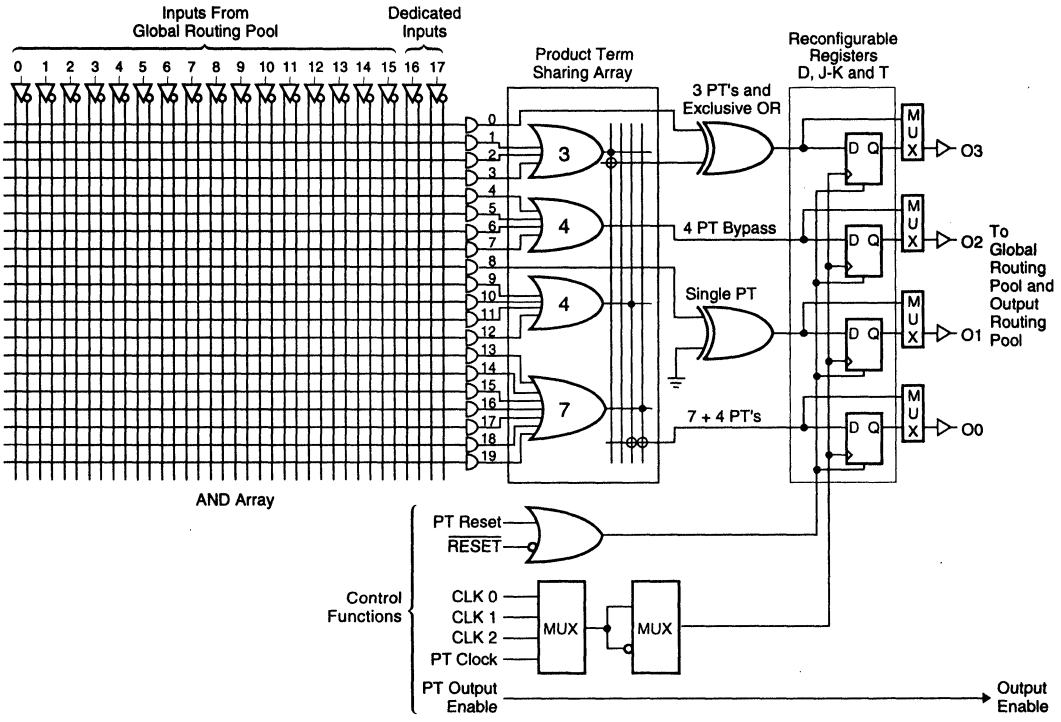
The Timing Model (see figure 4-19) shows all the internal characteristics listed in the following section. External switching characteristics are always the sum or difference of several internal characteristics. This section will first explain what each of the internal parameters are, then list each of the external parameters, and finally show which internal parameters they are derived from.

Figure 4-19. Timing Model for the pLSI and ispLSI Families



Internal Switching Characteristics

Figure 4-20. GLB Logic Diagram



Generic Logic Block (GLB) Timing

Parameters

Parameter #33, t_{4pt}
4 Product Term Delay

The time that it takes a signal to flow through the AND/OR/XOR gate circuitry in the GLB when it is configured in the 4 PT Bypass mode. It is measured from the input of the AND gates within the GLB to the D-input of the GLB Register, or to the output of the GLB when the register is bypassed.

Parameter #34, t_{xor20}
20 Product Term Delay

The time that it takes a signal to flow through the AND/OR/XOR gate circuitry in the GLB when it is configured in any of the remaining modes: the XOR gate mode and the sum of product terms mode.

The parameter is measured from the input of the AND gates within the GLB to the D-input of the GLB Register, or to the output of the GLB when the register is bypassed.

Parameter #35, t_{gbp}
GLB Register By-Pass Delay

The extra time that is added when the bypass path is used instead of the GLB register.

Parameter #36, t_{gfb}
GLB Feedback Delay

The GLB feedback delay is the time that it takes a signal to flow from a GLB output to the input of the GRP.

Parameter #37, t_{gco}
GLB Clock to Output Delay

The time that it takes after the rising edge of the GLB register clock for the GLB output to become valid.

Parameter #38, t_{gsu}
GLB Setup time before Clock

The time that the signal at the D-input of the GLB register must be stable before the rising edge of the GLB clock for the register to operate properly.

Parameter #39, t_{gh}
GLB Hold Time after Clock

The time that the signal at the D-input of the GLB register must remain stable after the rising edge of the GLB clock for the register to operate properly.

Parameter #40, t_{ggr}
GLB Global Reset Delay

The time that it takes a reset signal to propagate from the device dedicated Global Reset Pin to the Q output of the D-type flip-flop within the GLB.

Parameter #41, t_{gar}
GLB Asynchronous Reset Delay

The time that it takes an asynchronous reset signal generated using a product term within the GLB to reset the Q output of the GLB.

Input/Output Cell Timing Parameters

Parameter #42, t_{iat}
I/O Cell Latch Delay

The time that it takes a signal to flow through the I/O Cell Register/Latch when it is configured as a latch. This assumes that the Latch clock is held high throughout the entire time.

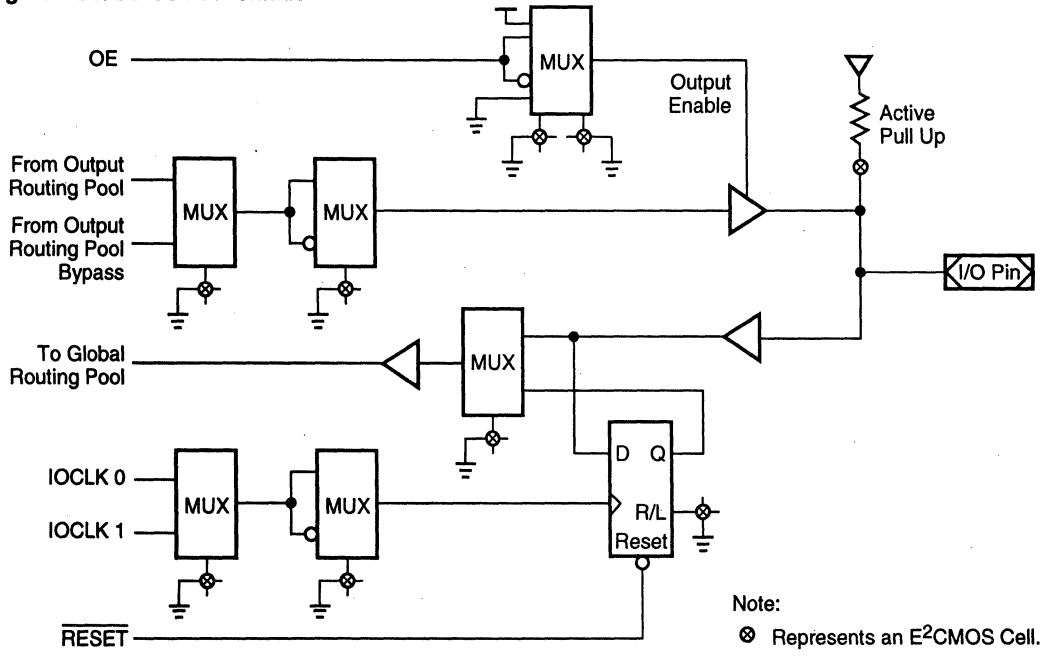
Parameter #43, t_{iobp}
I/O Cell Register By-Pass Delay

The extra time that is added when the bypass path is used instead of the Register/Latch.

Parameter #44, t_{iosu}
I/O Cell Setup before Clock

The time that the signal at the D-input of the I/O Cell register must be stable before the rising edge of the I/O clock for the register to operate properly. This parameter is only valid when the Register/Latch of the I/O Cell is configured in the Register Mode.

Figure 4-21. I/O Cell Schematic



Parameter #45, t_{ioh}
I/O Cell Hold Time after Clock

The time that the signal at the D-input of the I/O Cell register must remain stable after the rising edge of the I/O clock for the register to operate properly. This parameter is only valid when the Register/Latch of the I/O Cell is configured in the Register Mode.

Parameter #46, t_{ioco}
I/O Cell Clock to Output Delay

The time that it takes after the rising edge of the I/O Register clock for the I/O Cell Output to become valid.

Input and Output Delay Timing Parameters

Parameter #47, t_{din}
Dedicated Input Delay

The time that it takes a signal input on a dedicated input pin to propagate to a GLB AND gate input. Two Dedicated inputs are common to every GLB within a Megablock, and the t_{din} parameter will be the same for all GLBs within that megablock.

Parameter #48, t_{ib}
Input Buffer Delay

The time that it takes an external signal input on an I/O pin to propagate to the D-input of the I/O Cell Register/Latch, or to the I/O Cell Output when the Register/Latch is bypassed.

Parameter #49, t_{ob}
Output Buffer Delay

The time that a signal takes to flow through an I/O Cell when it is configured as an output.

Parameter #50, t_{mx}
Output Routing Pool Delay

The time that it takes a signal to flow through the Output Routing Pool when it is used in the normal (multiplexed) mode.

Parameter #51, t_{mxbp}
Output Routing Pool Bypass Delay

The time that it takes a signal to flow through the Output Routing Pool when it is used in the bypassed mode.

Clock and Enable Delay Parameters, GLB and I/O Cell

Parameter #52, t_{gy0}
Clock Delay, Y0 to GLB

The time that it takes for a clock signal which has been input on the Y0 Clock pin to reach the GLB register clock input.

Parameter #53, $t_{gy1/2}$
Clock Delay, Y1 or Y2 to GLB

The time that it takes for a clock signal which has been input on either the Y1 or the Y2 Clock pin to reach the GLB register clock input.

Parameter #54, t_{gpt}
Clock Delay, PT Clock to GLB

The time that it takes a clock signal generated using a product term within the GLB to propagate from the input of the GLB, through the AND gate to the clock input of the D-type register.

Parameter #55, t_{gcp}
Clock Delay, Clk GLB to GLB

The time that it takes a clock signal which has been created in the clock GLB to reach the clock input of a D-type register in any other GLB.

Parameter #56, t_{gen}
Enable Delay, GLB to I/O Cell

The time that it takes after a valid Output Enable signal has been generated in a GLB for an I/O Cell output buffer to go from 3-state to Active.

Parameter #57, t_{gdis}
Disable Delay, GLB to I/O Cell

The time that it takes after a valid Output Enable signal has been released in a GLB for an I/O Cell output buffer to go from Active to 3-state.

Parameter #58, t_{iocp}
Clock Delay, Clk GLB to I/O Cell

The time that it takes a clock signal which has been created in the clock GLB to reach the clock input of the Register/Latch in an I/O Cell.

Parameter #59, $t_{ioY2/3}$
Clock Delay, Y2 or Y3 to IO

The time that it takes for a clock signal which has been input on either the Y2 or the Y3 Clock pin to reach the clock input of the Register/Latch in an I/O Cell.

GRP Delays

Parameter #60, t_{grp4}
GRP Delay, Fanout 4

Parameter #61, t_{grp8}
GRP Delay, Fanout 8.

Parameter #62, t_{grp16}
GRP Delay, Fanout 16

Parameter #63, t_{grp32}
GRP Delay, Fanout 32

The only variable parameter in the pLSI and ispLSI devices is the propagation delay through the Global Routing Pool. The delay is dependent on how many GLB inputs the signal is driving. It can be approximated by referring to the graph for GRP delay provided in the datasheet of every device.

External Switching Characteristics

Note: Sometimes the specified external value and the sum of the internal values do not agree. This is because internal values are estimates only. These estimates are rounded off to the nearest nanosecond. The equations given are simply to explain what the factors are that make up an external parameter. For demonstration purposes the calculations below utilize pLSI 1032-80LJ specifications.

Parameter # 1, t_{pd1}
Data Propagation Delay, 4PT Bypass 15 ns.

t_{pd1} is the time data will take to propagate from any I/O pin configured as an input, through a single GLB configured in the 4 Product Term Bypass mode, to any I/O Pin configured as an output. The GRP routing delay is assumed to be four loads (average loading), and the ORP is configured in the bypass mode. This is the fastest path for getting a signal through the pLSI device.

t_{pd1} =#48+#43 +#60 +#33 +#35 +#51 +#49

t_{pd1} $=t_{ib}$ $+t_{iobp}$ $+t_{grp4}$ $+t_{4pt}$ $+t_{gbp}$ $+t_{mxbp}$ $+t_{ob}$

15 ns =2 +0 +3 +6 +0 +0 +4

Parameter # 2, t_{pd2}
Data Propagation Delay, ORP, 20PT 20 ns.

t_{pd2} is the time data will take to propagate from any I/O pin configured as an input, through a single GLB configured in the 20 Product Term mode, to any I/O Pin configured as an output. The GRP routing delay is assumed to be four loads (average loading), and the ORP is configured in the normal mode. This is a more common path for data in the pLSI device.

t_{pd2} =#48+#43 +#60 +#34 +#35 +#50 +#49

t_{pd2} $=t_{ib}$ + t_{iobp} $+t_{grp4}$ $+t_{xor20}$ $+t_{gbp}$ $+t_{mx}$ $+t_{ob}$

20 ns =2 +0 +3 +7.5 +0 +1 +4

Parameter #3, t_{co1}
External Clock to Output Delay, ORP Bypass 11 ns.

t_{co1} is the time it takes a signal to appear on an output pin after the rising of a clock applied to the Y0 Clock Pin. It assumes that the setup time for the data on the D-input of the GLB register has been met, and the ORP is configured in the bypass mode. t_{co1} is made up of two main components: the time it takes the clock to travel from the Y0 Clock Pin to the register clock input, and the sum of the register clock to output time and the signal propagation time from the register to the output pin.

t_{co1} =#52 + #37 + #51 + #49

t_{co1} $=t_{gy0}$ + t_{gco} + t_{mkbp} + t_{ob}

11 ns = 4 + 2 + 0 + 4

If either the Y1 or Y2 Clock Pin is used instead of Y0, Substitute $t_{gy1/2}$ for t_{gy0} in the equations above.

Parameter #4, t_{co2}
External Clock to Output Delay, ORP 14 ns.

t_{co2} is the time it takes a signal to appear on an output pin after the rising of a clock applied to the Y0 Clock Pin. It assumes that the setup time for the data on the D-input of the GLB register has been met, and the ORP is configured in the normal mode. t_{co2} is made up of two main components: the time it takes the clock to travel from the Y0 Clock Pin to the register clock input, and the sum of the register clock to output time and the signal propagation time from the register to the output pin.

t_{co2} =#52 + #37 + #50 + #49

t_{co2} $=t_{gy0}$ + t_{gco} + t_{mx} + t_{ob}

14 ns = 4 + 2 + 1 + 4

If either the Y1 or Y2 Clock Pin is used instead of Y0, Substitute $t_{GY\ 1/2}$ for t_{GY0} in the equations above.

Parameter #5, t_{co3}
Internal Synchronous Clock to Output Delay (GLB Clock Pin) 20 ns.

Internal Synchronous Clock parameters are measured using a synchronous clock generated in the clock GLB. (GLB C0 in the pLSI 1032) The state of that clock is controlled by a clock signal input on pin Y0. t_{co3} is the time it takes a signal to appear on an output pin after the rising edge of a clock on clock input pin Y0. It assumes that the setup time for the data on the D-input of the GLB register has been met, and that the ORP is configured in the normal mode.

t_{co3} =#52 + #37 + #55 + #37 + #50 + #49

t_{co3} $=t_{gy0}$ + t_{gco} + t_{gcp} + t_{gco} + t_{mx} + t_{ob}

20 ns = 4 + 2 + 4 + 2 + 1 + 4

Parameter #6, t_{co4}
Asynchronous Clock to Output Delay 20 ns.

t_{co4} is the time it takes a signal to appear on an output pin after the rising edge of a clock which has been generated using a product term within the GLB. It assumes that the setup time for the data on the D-input of the GLB register has been met, and the ORP is configured in the normal mode.

t_{co4} =#48+#43+#60 +#54 +#37 +#50 +#49

t_{co4} $=t_{ib}$ +t_{iobp}+t_{grp4} +t_{gpt} + t_{gco} + t_{mx} +t_{ob}

20 ns = 2 +0 +3 +5 +2 +1 +4

Parameter #7, t_{r1}
External Pin Reset to Output Delay 20 ns.

t_{r1} is the time it takes for a Reset signal applied to the device Reset pin to effect the outputs of all of the I/O Cells. The device is configured with the ORP in the normal mode.

t_{r1} = #40 + #50 + #49

t_{r1} = t_{ggr} + t_{mx} + t_{ob}

20 ns = 12 + 1 + 4

Parameter #8, t_{r2}
Asynchronous PT Reset to Output Delay 22 ns.

t_{r2} is the measurement of the time it takes a reset signal generated by a product term within a GLB to effect the output of an I/O Cell. The asynchronous reset effects only the outputs for a specific GLB, instead of every GLB on the device as parameter #38 does. t_{r2} is measured from the I/O Cell which is used to generate the reset input to the I/O cell output. It is specified using the 20 Product Term cell delay, and the ORP in normal mode.

t_{r2} =#48 + #43 +#60 +#41 +#50+#49

t_{r2} $=t_{ib}$ + t_{iobp} +t_{grp4} +t_{gar} +t_{mx} +t_{ob}

22 ns = 2 +0 +3 +9 +1 +4

Parameter #9, t_{en}
Input to Output Enable 20ns.

The time it takes to drive an I/O cell from 3-state (floating) to an active state. The qualifying input comes in on an I/O cell input, and the t_{en} parameter is measured from that input to the I/O cell output.

t_{en} =#48 +#43+#60 +#33 +#56

t_{en} $=t_{ib}$ +t_{iobp}+t_{grp4} +t_{4pt} +t_{gen}

20 ns = 2 +0 +3 +6 +7

Parameter #10, t_{dis}
Input to Output Disable 20ns.

The time it takes to drive an I/O cell from active state to 3-state (floating). The qualifying input comes from any I/O cell input, and the t_{dis} parameter is measured from that input to the I/O cell output.

t_{dis} =#48 + #43 + #60 + #33 + #57

t_{dis} = t_{ib} + t_{iobp} + t_{grp4} + t_{4pt} + t_{gdis}

20 ns = 2 + 0 + 3 + 6 + 7

External AC Recommended Operating Conditions

Parameter #11, **f_{max}**
Clock Frequency with Feedback 80 Mhz.

f_{max} is the highest operating frequency at which a pLSI part is guaranteed to operate when using internal feedback paths. The f_{max} parameter is calculated by configuring a 16 bit counter internal to the device, identifying the delay paths associated with that counter, and taking the reciprocal of that number. Register setup time and clock to output time also enter into the equation.

$$f_{max} = \frac{1}{\#37 + \#36 + \#60 + \#34 + \#38}$$

$$f_{max} = \frac{1}{t_{gco} + t_{gfb} + t_{grp4} + t_{xor20} + t_{gsu}}$$

$$80\ Mhz = \frac{1}{2 + 0 + 3 + 7.5 + 0}$$

Parameter #12, **$f_{max(External)}$**
Clock Frequency with External Feedback 50 Mhz.

f_{max} External is the highest operating frequency at which a pLSI 1032 device is guaranteed to operate when using external feedback. The f_{max} External parameter is calculated by configuring a counter which uses feedback paths external to the device, identifying the delay paths associated with that counter, and taking the reciprocal of that number. Register setup time and clock to output time also enter into the equation.

$$f_{maxext} = \frac{1}{\#37 + \#50 + \#49 + \#48 + \#43 + \#60 + \#34 + \#38}$$

$$f_{maxext} = \frac{1}{t_{gco} + t_{mx} + t_{ob} + t_{ib} + t_{iobp} + t_{grp4} + t_{xor20} + t_{gsu}}$$

$$50Mhz = \frac{1}{2 + 1 + 4 + 2 + 0 + 3 + 7.5 + 0}$$

Parameter #13, **t_{su1}**
Setup time before External Synchronous Clock, 4PT Bypass 9 ns.

t_{su1} is the time data (input on an I/O pin) must be stable before a clock (input on a dedicated clock pin) is asserted to assure correct operation of the register in the GLB. It is the difference of (the time it takes data to propagate from any I/O cell input pin) minus (the time that it takes a clock signal to propagate from any of the dedicated Clock Input Pins to the clock input of the D-type register) plus (the register setup time). t_{su1} is measured with the GLB configured in the 4 Product Term Bypass mode.

t_{su1} = (#48+#43 +#60 +#33) -(#52)+(#38)

t_{su1} = (t_{ib} +t_{iobp} +t_{grp4} + t_{4pt}) -(t_{gy0}) +(t_{gsu})

9 ns = (2 +0 +3 +6) -(4) +(0)

If either Y1 or Y2 are used as the synchronous clock input pin, then substitute $t_{gy1/2}$ for t_{gy0} in the equation above.

Parameter #14, **t_{su2}**
Setup time before External Synchronous Clock, 20PT 12 ns.

t_{su2} is the time data (input on an I/O pin) must be stable before a clock (input on a dedicated clock pin) is asserted to assure correct operation of the register in the GLB. It is the difference of (the time that it takes data to propagate from any I/O cell input pin) minus (the time that it takes a clock signal to propagate from any of the dedicated Clock Input Pins to the clock input of the D-type register) plus (the register setup time). t_{su2} is measured with the GLB configured in the 20 Product Term mode.

t_{su2} = (#48+#43+#60 +#34) -(#52)+(#38)

t_{su2} = (t_{ib} +t_{iobp}+t_{grp4} +t_{xor20}) -(t_{gy0}) +(t_{gsu})

12 ns = (2 +0 +3 +7.5) -(4) +(0)

If either Y1 or Y2 are used as the synchronous clock input pin, then substitute $t_{gy1/2}$ for t_{gy0} in the equation above.

Parameter #15, **t_{su3}**
Setup time before Internal Synchronous Clock 9 ns.

Internal Synchronous Clock parameters are measured using a synchronous clock generated in the Clock GLB (C0 in the pLSI 1032 Device). The state of that clock is controlled by a clock signal input on pin Y0. t_{su3} is the time data (input on an I/O pin) must be stable before a clock

(generated using the internal clock GLB as described) is asserted to assure correct operation of the register in the GLB. It is the difference of (the time it takes data to propagate from any I/O cell input pin) minus (the time that it takes a clock signal to propagate from clock pin Y0 through the dedicated clock GLB to the clock input of the D-type register) plus (the register setup time). t_{su3} is measured with the GLB configured in the 20 Product Term mode.

t_{su3} = (#48+#43+#60 +#34) - (#52+#37+#55)+ (#38)

t_{su3} = $(t_{ib}+t_{iobp}+t_{grp4}+t_{xor20})-(t_{gy0}+t_{gco}+t_{gcp})+(t_{gsu})$

9 ns = (2 +0 +3 +7.5)-(4 +2 +4) +(0)

Parameter #16, **t_{su4}**
Setup time before Asynchronous Clock 9 ns.

t_{su4} is the time data (input on an I/O pin) must be stable before a clock (input on an I/O pin) is asserted to assure correct operation of the register in the GLB. It is the difference of (the time it takes data to propagate from any I/O cell input pin to the D-input of the GLB register) minus (the time it takes a clock signal to propagate from an I/O cell configured as an asynchronous clock input to the clock input of the GLB register) plus (the register setup time).

t_{su4}= (#48+#43+#60+#34) - (#48+#43+#60 +#54)+(#38)

t_{su4}= $(t_{ib}+t_{iobp}+t_{grp4}+t_{xor20})-(t_{ib}+t_{iobp}+t_{grp4}+t_{gpt})+(t_{gsu})$

9 ns= (2 +0 +3 +7.5)-(2 +0 +3 +5) +(0)

Parameter #17, **t_{h1}**
Hold time after External Synchronous Clock 2 ns.

t_{h1} is the time data (input on an I/O pin) must be held stable after a clock (input on a dedicated clock pin) is asserted to assure correct operation of the register in the GLB. It is the difference of (the time it takes a clock signal to propagate from any of Clock Input Pins Y0, Y1 or Y2 to the clock input of the D-type register) minus (the time it takes data to propagate from any I/O cell input pin to the D-input of the GLB register) plus (the register hold time). Because the answer was a negative number, the parameter was set to zero. tH1 corresponds to both setup times, t_{su1} and t_{su2}.

t_{h1} = (#52) - (#48 +#43 +#60+#33) +(#39)

t_{h1} = $(t_{gy0})-(t_{ib}+t_{iobp}+t_{grp4}+t_{4ptmin})+(t_{gh})$

2 ns = (4) -(2 +0 +3 +6) +(2)

If either Y1 or Y2 are used as the synchronous clock input pin, then substitute $t_{gy1/2}$ for t_{gy0} in the equation above.

Parameter #19, **t_{h3}**
Hold time after Internal Synchronous Clock 8 ns.

t_{h3} is the time data (input on an I/O pin) must be held stable after a clock (generated using the internal clock GLB) is asserted to assure correct operation of the register in the GLB. It is the difference of (the time that it takes a clock signal to propagate from the Clock GLB to the clock input of the D-type register) minus (the time that it takes data to propagate from any I/O cell input pin to the D-input of the GLB register) plus (the register hold time).

t_{h3} = (#52+#37+#55) - (#48+#43+#60+#33) +(#39)

t_{h3} = $(t_{gy0}+t_{gco}+t_{gcp})-(t_{ib}+t_{iobp}+t_{grp4}+t_{4ptmin})+(t_{gh})$

8 ns = (4 +2 + 4) -(2 +0 + 3 +6) +(2)

Parameter #20, **t_{h4}**
Hold time after Asynchronous Clock 8 ns.

t_{h4} is the time data (input on an I/O pin) must be held stable after a clock (input on an I/O pin) is asserted to assure correct operation of the register in the GLB. It is the difference of (the time that it takes a clock signal to propagate from any the Clock GLB to the clock input of the D-type register) minus (the time that it takes data to propagate from any I/O cell input pin to the D-input of the GLB register) plus (the register hold time).

t_{h3} = (#48+#43+#60+#54) - (#48+#43+#60+#33) +(#39)

t_{h3} = $(t_{ib}+t_{iobp}+t_{grp4}+t_{gpt})-(t_{ib}+t_{iobp}+t_{grp4}+t_{4ptmin})+(t_{gh})$

8 ns = (2 +0 +3 +5) -(2 +0 +3 +6) +(2)

Parameter #21, **t_{rw1}**
External Reset Pulse Duration 10 ns.

t_{rw1} is the minimum time a reset pulse can be applied to the external Reset Pin for it to be correctly recognized .

Parameter #22, **t_{rw2}**
Asynchronous Reset Pulse Duration 10 ns.

t_{rw2} is the minimum time a reset pulse can be applied to an I/O Pin for it to be correctly recognized. This parameter applies to the asynchronous Product Term Reset within a GLB.

Parameter #23, **t_{wh1}**
External Synchronous Clock Pulse Duration, High 6 ns.

Parameter #24, **t_{wl1}**
External Synchronous Clock Pulse Duration, Low 6 ns.

t_{wh1} and t_{wl1} are the minimum high and low duration for a GLB clock applied to any of the three external Clock Pins.

Parameter #25, t_{wh2}
Asynchronous Clock Pulse Duration, High 6 ns.

Parameter #26, t_{wl2}
Asynchronous Clock Pulse Duration, Low 6 ns.

t_{wh2} and t_{wl2} are the minimum high and low duration for a clock applied to any I/O Cell pin. This parameter applies to the asynchronous clock used within a GLB.

Operating Conditions if using I/O Cell Register

Parameter #27, t_{su5}
Setup time before External Synchronous Clock 0 ns.

t_{su5} is the time data (input on an I/O pin) must be stable before a clock (input on a dedicated clock pin) is asserted to assure correct operation of the register in the I/O Cell. It is the difference of (the time that it takes data to propagate to the D-input of the Register/Latch in the I/O Cell) minus (the time that it takes a clock signal to propagate from any of the dedicated Clock Input Pins to the clock input of the register) plus (the register setup time).

t_{su5} = (#48) - (#59) + (#44)

t_{su5} = (t_{ib}) - ($t_{ioy2/3}$) + (t_{Iosu})

0 ns = (2) - (4) + (0)

Parameter #28, t_{su6}
Setup time before Internal Synchronous Clock (GLB CLK Pin) 0 Ns.

t_{su6} is the time data (input on an I/O pin) must be stable before a clock (generated using the internal clock GLB) is asserted to assure correct operation of the register in the I/O Cell. It is the difference of (the time it takes data to propagate to the D-input of the Register/Latch in the I/O Cell) minus (the time it takes a clock signal to propagate from the dedicated clock GLB to the clock input of the D-type register) plus (the register setup time). t_{su6} is measured with the GLB configured in the 20 Product Term mode. (Because the answer is a negative number, the parameter was set to zero.)

t_{su6} = (#48) - (#48+ #43 + #60 + #58) + (#44)

t_{su6} = (t_{ib}) - (t_{ib} + t_{iobp} + t_{grp4} + t_{iocp}) + (t_{iosu})

0 ns = (2) - (2 + 0 + 3 + 4) + (0)

Parameter #29, t_{h5}
Hold time after External Synchronous Clock 8 ns.

t_{h5} is the time data (input on an I/O pin) must be held stable after a clock (input on a dedicated I/O clock pin) is asserted to assure correct operation of the register in the I/O Cell. It is the difference of (the time it takes a clock signal to propagate from either of the dedicated I/O Clock Input Pins Y2 and Y3 to the clock input of the D-type register) minus (the time it takes data to propagate to the D-input of the Register/Latch in the I/O Cell) plus (the register hold time) .

t_{h5} = (#59) - (#48) + (#45)

t_{h5} = ($t_{ioy2/3}$) - (t_{ib}) + (t_{ioh})

8 ns = (4) - (2) + (1)

Parameter #30, t_{h6}
Hold time after Internal Synchronous Clock 15 Ns.

Internal Synchronous Clock parameters are measured using a synchronous clock generated in GLB C0. The state of that clock is controlled by a clock signal input on pin Y0. t_{H6} is the time data (input on an I/O pin) must be held stable after a clock (generated using the internal clock GLB as described) is asserted to assure correct operation of the register in the I/O Cell. It is the difference of (the time it takes a clock signal to propagate from any the Clock GLB to the clock input of the D-type register/ Latch) minus (the time it takes data to propagate from any I/O cell input pin to the D-input of the I/O Cell register) plus (the register hold time.)

t_{h6} = (#52+#37+ #58) - (#48) + (#45)

t_{h6} = (t_{gy0} + t_{gco} + t_{iocp}) - (t_{ib}) + (t_{ioh})

15 ns = (4 +2 +4) - (2) + (1)

Parameter #31, t_{wh3}
Clock Pulse Duration, High 6 Ns.

Parameter #32, t_{wl3}
Clock Pulse Duration, Low 6 Ns.

t_{wh3} and t_{wl3} are the minimum high and low duration for an I/O Cell clock applied to either of the two external I/O Clock Pins (Y2 or Y3), or generated in the clock GLB. (GLB C0 in the pLSI 1032 device.)

ispLSI Programming Information

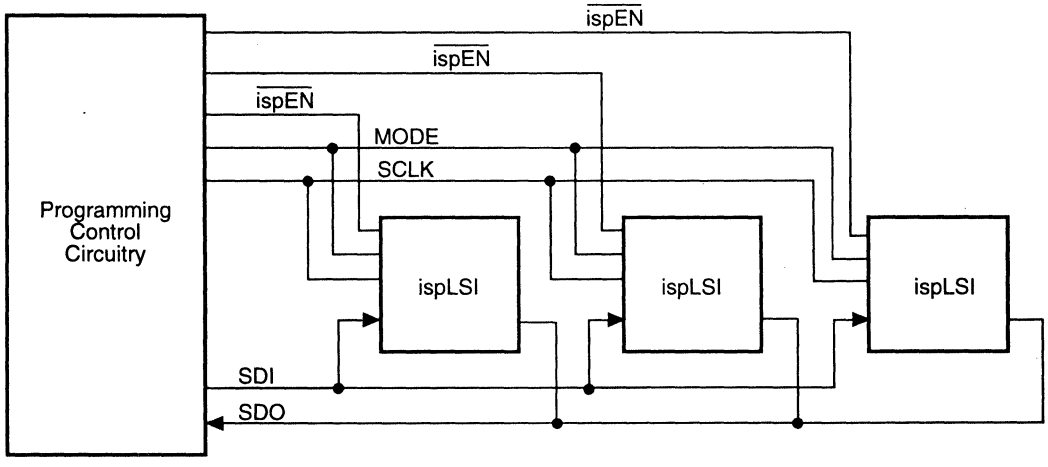
The following general programming information on the ispLSI (in-system programmable Large Scale Integration) devices describes how the internal state machine is implemented for programming and how to use the five programming interface signals to step through the state machine. The device specific information, such as timing and pin outs, can be found in the individual data sheets. The programming information given in this section applies to all ispLSI devices.

Programming Overview

To distinguish between normal operation and programming, two modes are defined: normal mode and edit mode. Once the device is in edit mode, the entire programming operation of the device is controlled by the internal isp state machine. The in-system programming ENable ($\overline{\text{ispEN}}$) signal controls the device operation modes.

The programming is controlled by the on-chip state machine via five programming interface signals. The $\overline{\text{ispEN}}$ signal is used to enable and disable the four programming control signals which include Serial Data In (SDI), Mode control (MODE), Serial Data Out (SDO) and Serial Clock (SCLK) signals. Figure 4-22 illustrates one such possible configuration for programming multiple pLSI devices. With this scheme the $\overline{\text{ispEN}}$ signal for individual devices is enabled separately and one device is placed in the edit mode at a time. Since the other devices are in the normal mode, they can continue to perform normal system functions. When the device is in normal mode, the four programming control signal pins can be used as normal Dedicated Input Pins. This simple scheme requires connecting all four programming control signal pins together and precludes their use as dedicated inputs for normal system functions. $\overline{\text{ispEN}}$ is the only programming interface signal that is dedicated to a pin.

Figure 4-22. isp Programming Loop



Normal Mode

In normal mode the four programming control pins become Dedicated Input pins. By multiplexing the programming control pins, these programming control pins can have a normal input function during normal mode. Figure 4-23 and 4-24 illustrate two alternate schemes which allow the designer to utilize the four programming control signal pins for performing normal system functions. Internal to the device, the programming functions are completely isolated from the normal operating functions when the device is in normal mode. Keeping the $\overline{\text{ispEN}}$ signal high puts the device in normal mode. For simplicity, the four programming control pins can be left unused for normal input functions. By leaving these pins unused, the programming interface is made easier where the programming signals and normal mode input signals need not be multiplexed.

Edit Mode

Programming circuitry is enabled by driving the $\overline{\text{ispEN}}$ signal low which puts the device in edit mode. In edit mode, all the functional I/O pins and input pins that are not used during programming are 3-stated. With the exception of the SDO signal, the remainder of the programming interface signals are input signals. When multiplexing the programming interface signals, the input driving the SDO pin must be 3-stated to make sure that there is no signal contention. All programming is accomplished in the edit mode by controlling the programming state machine with the MODE and SDI and using the SCLK to clock programming data in and out through SDI and SDO pins. SDI has a dual role as one of the two control signals for the state machine and as the serial data input. To avoid any internal register data contentions, Lattice recommends that the device reset pin be pulled to ground when the device is in edit mode.

Programming Interface

The five programming interface pins are $\overline{\text{ispEN}}$, SDI, MODE, SDO and SCLK. Once in edit mode, programming is controlled by SDI, MODE, SDO and SCLK signals. In normal mode, the programming control pins can be used as dedicated inputs to the device.

$\overline{\text{ispEN}}$ is an active low, dedicated enable pin, which enables the four programming control pins when it is driven low (V_{IL}) and disables the programming control pins when it is driven high (V_{IH}). All other I/O pins are 3-stated during edit mode and pulled up by the internal active pull-up resistors (equivalent to 100KΩ).

SDI performs two different functions. First, as the input to the serial shift register and second, as one of the two control pins for the programming state machine. Because of this dual role, SDI's function is controlled by the MODE signal. When MODE is low SDI is the serial input to the shift registers and when MODE is high SDI becomes the control signal. Internal to the device, the SDI is multiplexed to address shift register, high order data shift register and low order data shift register. The different shift instructions of the state machine determine which of these shift registers gets the input of the SDI.

The MODE signal combined with the SDI signal controls the programming state machine. This signal connects in parallel to all ispLSI devices.

SCLK is the serial shift register clock that is used to clock the internal serial shift registers. A low-to-high (positive) clock transition clocks the state machine. It also connects in parallel to all ispLSI devices. Similar to SDI, the shift instructions determine which of the shift registers are clocked for the data input from SDI.

SDO is the output of the serial shift registers. The selection of shift register is determined by the state machine's shift instruction. In the flow through instruction and when MODE is driven high, the SDO connects directly to the SDI, and bypasses the device's shift registers. Since this is the only output pin for the edit mode, this signal will drive the external devices that are connected to this pin.

Programming Details

The programming is completely controlled by the state machine, once the device is in the edit mode. The state machine consists of three states, in which all programming related operations are performed. In order to run these programming operations, five bit instructions are defined (see table 4-8). Each instruction is then shifted into the device in one of the three states and executed in another state. The initial state of the state machine is used when the device is idle during edit, or to shift out the eight bit device identification code.

The following sections describe the general information about the critical timing parameters, state machine, state machine instructions, and device layout that apply to all the ispLSI devices. Any device specific information like the size of the shift registers and the device specific timing information can be found in the individual device data sheet.

Figure 4-23. The Scan and Multiplex Programming Mode

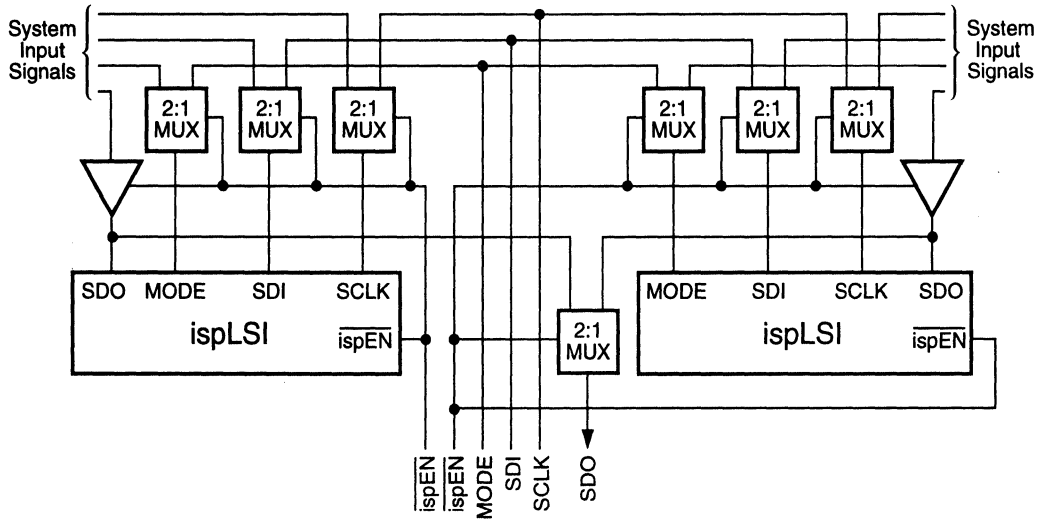
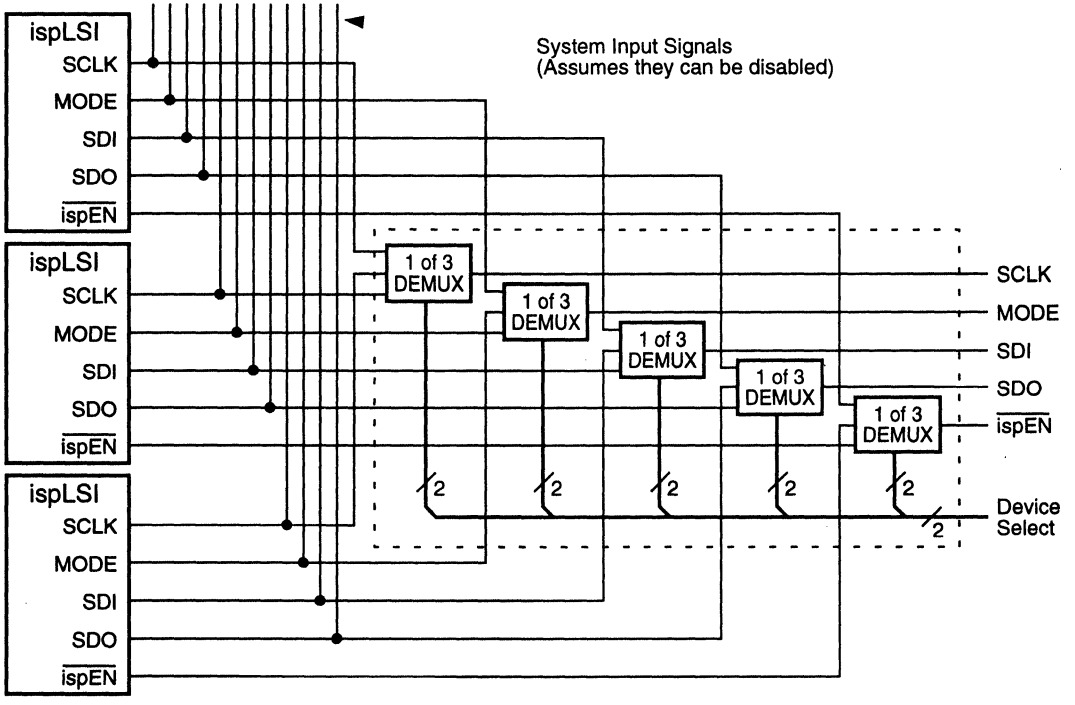


Figure 4-24. The Scan and Multiplex Programming Mode



Critical Timing Parameters

When programming the ispLSI devices there are several critical timing parameters that must be met in order to program the devices properly. The most critical of these parameters are the programming pulse width (t_{pwp}) and the bulk erase pulse width (t_{bew}). These pulse widths determine the programming and erasing of the E^2 cells. Figure 4-25 shows these critical program and erase timing specifications.

Along with the two programming and erasing specification, the following timing specifications must also be met.

t_d - Time delay that must elapse between events. It is the time delay from the termination of the previous event.

t_{isp} - Specifies the time it takes to get into the isp mode after $\overline{ispEN}$ signal is activated or the time it takes to come out from the isp mode after the $\overline{ispEN}$ becomes inactive.

t_{su} - Set up time of the control signals before the SCLK or the set up time of input signals against other control signal where applicable (see figure 4-17).

t_h - Hold time of the control signal after the SCLK. It also applies to the same input signals from the set up time.

t_{clk} - Minimum clock pulse width high time for SCLK.

t_{pwv}- Verify or read pulse width. The minimum time requirement from the rising clock edge of verify/ load instruction execution to the next rising clock edge (see figure 4-25).

t_{rst} - Power on reset timing requirement. t_{rst} must elapse after power up before any operations are performed on the device.

All the programming timing parameters are summarized in the timing diagram (see figure 4-26).

Figure 4-25. Program, Verify & Bulk Erase Timing

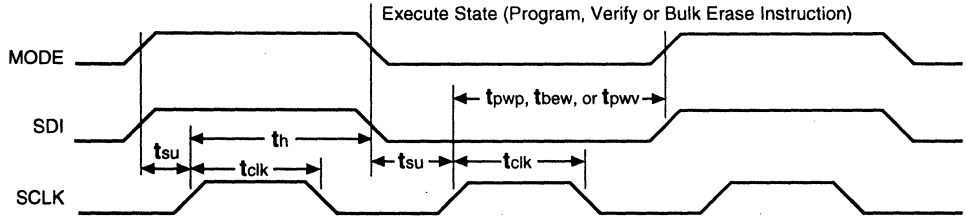


Figure 4-26. isp Programming Timing Requirements

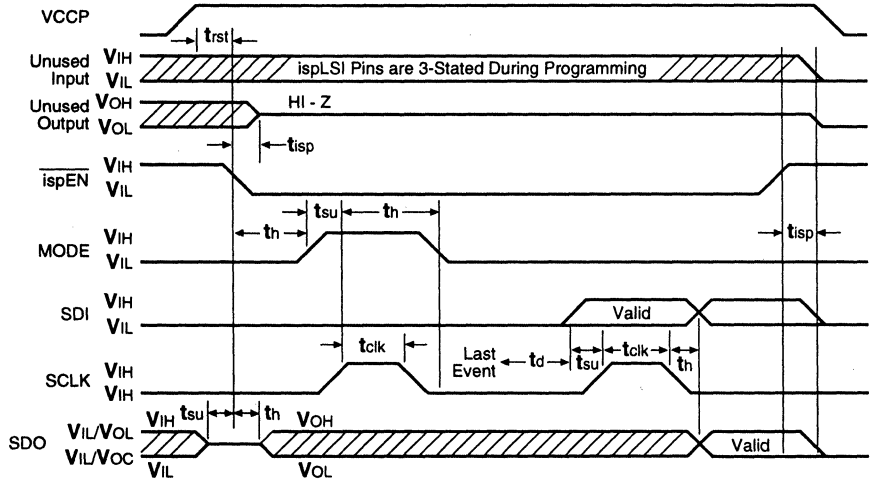
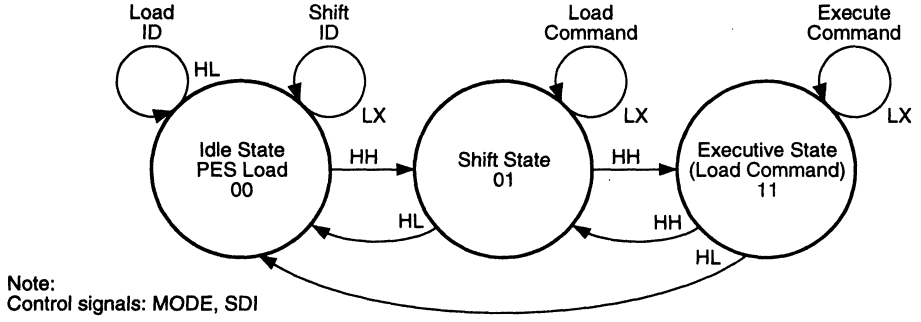


Figure 4-27. Programming State Machine



State Machine Operation

The state machine has three states to control the programming and uses the MODE and SDI as inputs for each state. Based on these input signals, each of the three states make decisions to either stay in the same state or to branch to another state. The three states are Idle/ID State, Command Shift State and Execute State. The programming state machine diagram below shows the three states and the status of the control signals in each state for indicated operation.

Idle/ID State

The Idle/ID state is the first state which is active when the device gets into the edit mode. The state machine is in the Idle/ID state when the device is idle, in the edit mode, or when the user needs to read the device identification. The eight bit device identification is loaded into the shift register by driving MODE high, SDI low and clocking the state machine with SCLK. Once the ID is loaded, it is read out serially by driving MODE low. Notice that when reading the device ID serially, SDI can either be high or low (don't care) and the state machine needs only seven clocks to read out eight bits of ID. The default state for the control signals is MODE high and SDI low. State transition to Command Shift State occurs when both MODE and SDI are high while state machine gets a clock transition. The following table lists the eight bit device ID's for all the ispLSI devices. As with most shift registers the Least Significant Bit (LSB) of the ID gets shifted out from the SDO first.

Command Shift State

This state is strictly used for shifting in the command instructions into the state machine. The entire five-bit instruction set is listed in the next section. When MODE is low and SDI is don't care in the Command Shift State,

Table 4-7. ispLSI Device ID Codes

Device	MSB	LSB
ispLSI 1016	0000	0001
ispLSI 1024	0000	0010
ispLSI 1032	0000	0011
ispLSI 1048	0000	0100

SCLK shifts the instruction into the state machine. Once the instruction is shifted into the state machine, the state machine must transition to the Execute State to execute the instruction. Driving both MODE and SDI high and applying the clock will transfer the state machine from Command Shift State to Execute State. If needed, the state machine can move from Command Shift State to Idle/ID State by driving MODE high and SDI low.

Execute State

In the Execute State, the state machine executes instructions that are loaded into the device in the Command Shift State. For some instructions, the state machine requires more that one clock to execute the command. An example of this multiple clock requirement is the address or data shift instruction. The number of clock pulses required for these instructions depends on the device shift register sizes (refer to the isp programming section of the data sheet). When executing instructions such as Program, Verify or Bulk Erase, the necessary timing requirements must be followed to make sure that the commands are executed properly. For specific timing information refer to the individual data sheets.

To execute a command, the MODE is driven low and SDI is don't care. For multiple clock instructions these control signal's status must remain the same throughout the

duration of the execution. MODE high and SDI high will take the state machine back to the Command Shift State and MODE high and SDI low will take the state machine to the Idle/ID State.

Instructions

Table 4-8 lists the instructions that can be loaded into the state machine in the Command Shift State and then executed in the Execute State. Notice that reading the device identification is done during the Idle/ID State and does not require an instruction.

Table 4-8. State Machine Instruction Set

Instruction	Operation	Description
00000	NOP	No operation performed.
00001	ADDSHIFT	Address Register Shift: Shift address into the address shift register from SDI.
00010	DATASHIFT	Data Register Shift: Shifts data into or out of the data serial register.
00011	UBE	User Bulk Erase: Erase the entire device.
00100	GRPBE	Global Routing Pool Bulk Erase: Bulk eráses the GRP array only.
00101	GLBBE	Generic Logic Block Bulk Erase: Bulk erases all the GLB array only.
00110	ARCHBE	Architecture Bulk Erase: Bulk erases the architecture array and I/O configuration only.
00111	PRGMH	Program High Order Bits: The data in the shift register is programmed into the addressed row's high order bits.
01000	PRGML	Program Low Order Bits: The data in the data shift register is programmed into the addressed row's low order bits.
01001	PRGMSC	Program Secirity Cell: Programs the security cell of the device.
01010	VER/LDH	Verify/Load High Order Bits: Load the data from the selected row's high order bits into the data shift register for programmed verification.
01011	VER/LDL	Verify/Load Low Order Bits: Load the data from the selected row's low order bits into the data shift register for programmed verification.
01100	GLBRLD	Generic Logic Block Preload: Preloads the rigisters in the GLBs with the data from SDI. All registers in the GLB from a serial shift register. Refer to device layout section for details.
01101	IOPRLD	I/O Preload: Preloads the I/O registers with the data from SDI. All registers in the I/O cell form a serial shift register (the same order as GLB registers).
01110	FLOWTHRU	Flow Through: Bypasses all the internal shift registers and SDO becomes the same as SDI.
10010	VE/LDH	Verify Erase/Load High Order Bits: Load the data from the selected row's high order bits into the data shift register for erased verification.
10011	VE/LDL	Verify Erase/Load Low Order Bits: Load the data from the selected row's low order bits into the data shift register for erased verification.

While it is possible to erase the individual arrays of the device, it is recommended that the entire device be erased (User Bulk Erase) and programmed in one operation. This Bulk Erase operation should precede every programming cycle as an initialization.

When a device is secured by programming the security cell (PRGMSC), the on-chip verify and load circuitry is disabled. Securing of the device should be done as the last procedure after all the device verifications have been completed. The only way to erase the security cell is to perform a bulk erase on the device.

Device Layout

The purpose of knowing the device layout is to be able to translate the JEDEC format programming file into the serial data stream format for programming ispLSI devices. Two main factors determine how the translation is implemented. The length of the address shift register and the length of the data shift register. The length of the address shift register indicates how many rows of data are to be programmed into the device. The length of the data shift register indicates how many bits are to be programmed in each row. Both registers operates on the First In First Out (FIFO) basis where the Least Significant Bit (LSB) of the data or address is shifted in first and the Most Significant Bit (MSB) of the data or address is shifted in last. For the data shift register, the low order bits and the high order bits are separately shifted.

Figure 4-28 illustrates the general layout of all the ispLSI devices. Symmetric to the center row and starting from the two outermost rows of the device, there are two sets of I/O and architecture arrays, two sets of GLB arrays and one set of GRP array.

Between all ispLSI devices there are exactly the same number of rows for the I/O and architecture array as there are for the GLB array. The GRP array size is proportional to the size of the device. According to the size of the GRP array, the size of the address shift register gets adjusted for different devices. Tables 4-9 and 4-10 summarize the array and shift register sizes for all ispLSI devices.

Using ispLSI 1032 as a specific example the transition from a JEDEC format programming file to the ispLSI device format is illustrated below. The JEDEC format programming file for ispLSI 1032 is organized as follows:

```
L00000   01010101....  ....010101010*
L00040   11111111....  ....111111111*
L00080   00000000....  ....000000000*
L00120   11111111....  ....111111111*
L00160   01010101....  ....010101010*
   •
   •
   •
```

The L field in the JEDEC programming file indicates the first cell number of each row. The JEDEC standard requires that there is at least the beginning cell number L00000. L fields of the subsequent lines are optional. From this reference cell location all other cell locations can be determined. Zero in the cell location indicates that the E^2 cell in that particular location is programmed (or has a logic connection equivalent to a metal fuse being intact). A one (1) in the cell location indicates that the cell is erased (equivalent to blown fuse). The successful operation of Fusemap in the Lattice software generates this JEDEC standard programming file. This is also the standard that is widely used in the PLD industry for translating design ideas into the device specific layout for programming.

Table 4-9. Summary of address Shift Register and Array Sizes

Address SR Rows	ispLSI 1016	ispLSI 1024	ispLSI 1032	ispLSI 1048
2 - I/O & Arch. Array	12	12	12	12
2 - GLB Array	72	72	72	72
GRP Array	12	18	24	36
Total Address SR Rows	96	102	108	120

Figure 4-28. ispLSI Device & Shift Register Layout

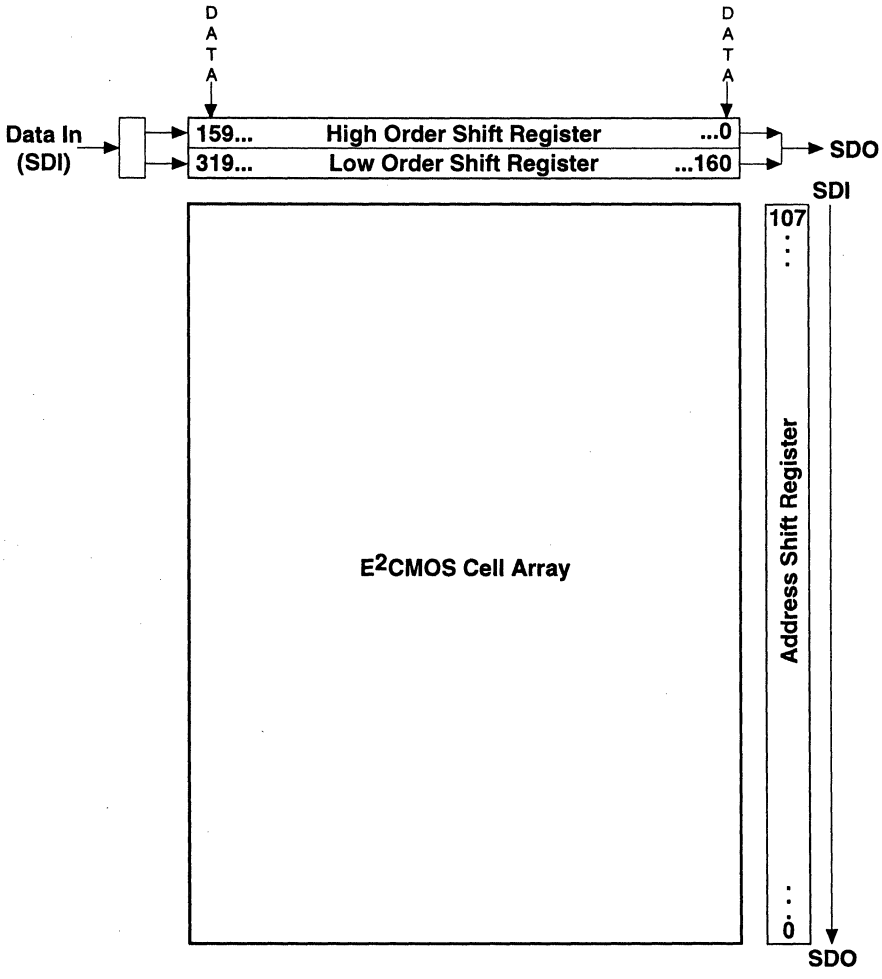


Table 4-10. Summary of Data Shift Register

	ispLSI 1016	ispLSI 1024	ispLSI 1032	ispLSI 1048
High Order Data SR LSB	0	0	0	0
High Order Data SR MSB	79	119	159	239
Low Order Data SR LSB	80	120	160	240
Low Order Data SR MSB	159	239	319	479
Data SR Size (Bits)	160	240	320	480

Translating the JEDEC programming file into the ispLSI 1032 device format the layout should be as follows:

		LSB	MSB
Row#	0000 High	L00000....	....L00159
Row#	0000 Low	L00160....	....L00319
Row#	0001 High	L00320....	....L00479
Row#	0001 Low	L00480....	....L00639

•
•
•

The least significant bit of the data shift register matches up with the lowest cell number of the corresponding cells from the JEDEC programming file.

Command Stream

The first step of programming the ispLSI devices is to determine type of devices. This can be done by reading the eight-bit device ID of all the devices. By keeping the SDI to a known level (either high or low), the ID shift can be terminated when a sequence of eight ones or eight zeros are read. From the device ID the serial bit stream for programming can be arranged. A typical programming sequence is as follows:

1) ADDSHFT command shift
2) Execute ADDSHFT command
3) Shift address
4) DATASHFT command shift
5) Execute DATASHFT command
6) Shift high order data
7) PRGMH command shift
8) Execute PRGMH
9) DATASHFT command shift
10) Execute DATASHFT command
11) Shift low order data
12) PRGML command shift
13) Execute PRGML
14) Repeat from 1) until all rows are programmed.

Diagnostic

One of the diagnostic features of the ispLSI devices is the register preload. Both GLB registers and I/O cell registers become serial shift registers during the corresponding register preload instruction execution. Data can either be shifted into or out of these shift registers for system diagnostic functions. The order in which these registers form the shift register is summarized in table 4-11.

There are some restrictions on the GLB and I/O clocks that must be followed in order to properly use the register preload features. Use the non-inverted GLB clocks and keep the GLB clock signal high before going into the preload (edit) mode. Similarly, the inverted I/O clocks must be used and it must be driven low before going into the preload mode.

Table 4-11. GLB and I/O Preload Shift Registers

	ispLSI 1016	ispLSI 1024	ispLSI 1032	ispLSI 1048
GLB Registers	A7..A0,B0..B7	B3..B0,A7..A0, B4..B7,C0..C7	B7..B0,A7..A0, C0..C7,D0..D7	C7..C0,B7..B0, A7..A0,D0..D7 E0..E7,F0..F7
I/O Registers	I/O15..I/O0, I/O16..I/O31	I/O23..I/O0, I/O24..I/O47	I/O31..I/O0, I/O32..I/O62	I/O47..I/O0, I/O48..I/O79

A different type of diagnostic feature takes advantage of the in-system programmability. For diagnostics, the isp devices can be programmed with the test functions in diagnostic mode. When the diagnostic is complete, the isp devices can then be reprogrammed with the functional pattern.

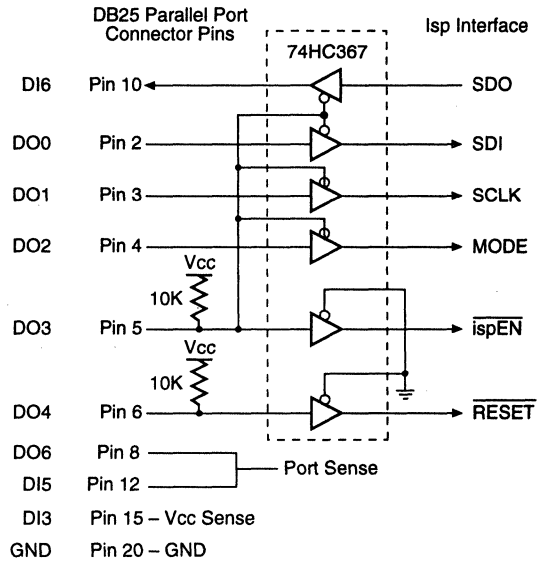
Programming Tools

As part of the engineering serial programming C program, some C language routines are available from Lattice to provide examples of the following. The routines use the PC parallel port to interface with the ispLSI device and program one ispLSI device at a time.

- ❑ Programming a Device.
- ❑ Verifying a Device.
- ❑ Reading a Device.
- ❑ Securing a Device.
- ❑ Bulk Erasing a Device.

These five routines are included in the source code listing. Subroutines that are necessary to run these routines are also given in the listing. Note that these routines were specifically developed to program the ispLSI 1032. The timing loops in the routines must be adjusted according to the speed of the PC. The parallel port connection, in figure 4-29, illustrates how the pins of the parallel port are connected to the isp interface. Similar circuitry can be built into the system to implement the in-system programming.

Figure 4-29. PC Parallel Port Connection



C Source Code Listing

```
/*********************************************************************/
/**    Rev                                                          **/
/**    1.0                                                          **/
/**       This program was translated from ispLSI 1032 BASIC        **/
/**    2.1                                                          **/
/**                                                                 **/
/*********************************************************************/
#include  <stdio.h>
#include  <conio.h>
#include  <graph.h>
#include  <dos.h>
#include  <time.h>
#include  <string.h>
/* Global Variables */
/* max = highest row address */
int max = 108;
/* bit = number of bits in a row of file data */
int bit = 160;
unsigned int inport = 0x0279;
unsigned int outport = 0x0278;
/*********************************************************************/
/*                                                                   */
/********************  PROGRAM THE ispLSI 1032  **********************/
/*                                                                   */
/*********************************************************************/
   Program(flg1)
   int flg1;
   { char ansr,flg2;
      long int i,j,k,ii; _clearscreen(_GCLEARSCREEN);
      M3:
      _settextcolor(11);
      sprintf(buffer,"ENTER NAME OF FILE TO DOWNLOAD\n");
      _outtext(buffer);
      _settextcolor(oldfgd);
      scanf("%s",name);
      if ((fpr = fopen(name,"r"))==NULL)
         {_clearscreen(_GCLEARSCREEN);
         _settextcolor(12);
         sprintf(buffer,"%s DOES NOT EXIST\n",name);
         _outtext(buffer); sprintf(buffer,"DO YOU WANT TO CONTINUE ?  Y or N \n");
         _outtext(buffer);
         _settextcolor(oldfgd); ansr=toupper(getch());
         if (ansr=='Y')goto M3;
            else goto abort1;
         }
      line[0]='a'; line[1]='a';
      i=0;
      while(((line[0]!='*')||(line[1]!='L'))&&(i<20))
      {ReadLine(fpr);
      i++;}
```

```
   if(i>18){_clearscreen(_GCLEARSCREEN);
         _settextcolor(12);
         sprintf(buffer,"%s  IS  NOT  IN  STANDARD  FORMAT  !!!\n",name);
         _outtext(buffer);
         sprintf(buffer,"DO YOU WANT TO CONTINUE ?   Y or N \n");
         _outtext(buffer);
         _settextcolor(oldfgd);
         ansr=toupper(getch());
         if  (ansr=='Y')goto  M3;
         else  goto  abort1;}
   i=0;
   while  ((infile[i]=fgetc(fpr))!='*')
      {  if  (infile[i]=='\n');
      else  i++;
      }

ii=0;
cont1:
if  (( 8  &  inp(inport))==0)
   {ii++;
   _clearscreen(_GCLEARSCREEN);
   _settextcolor(12);
   sprintf(buffer,"SUPPLY  POWER  IS  OFF,  PLEASE  TURN  IT  ON  ");
   _outtext(buffer);
   for(i=0;i<ii;i++){sprintf(buffer,"!");
               _outtext(buffer);}
   sprintf(buffer,"\nDO  YOU  WANT  TO  CONTINUE  ?   Y or N \n");
   _outtext(buffer);
   _settextcolor(oldfgd);
   ansr=toupper(getch());
   if  (ansr=='Y')goto  cont1;
   else  goto  abort1;}

   Bulk_erase();
   ptr2=0;
   for  (adrs=1;  adrs<max+1;  adrs++)
      {  _clearscreen(_GCLEARSCREEN);
      _settextcolor(15);
      sprintf(buffer,"STANDBY  PROGRAMMING  IN  PROGRESS  %d\n",max+1-adrs);
      _outtext(buffer);
      Addr_gen();
      Data_SDI();
      Prog_even();
      Execute();
      Prog_delay();
      Execute();
      Data_SDI();
      Prog_odd();
      Execute();
      Prog_delay();
      Execute();
   }
   if  (flg1==0){_clearscreen(_GCLEARSCREEN);
```

```
   sprintf(buffer,"***  DONE  PROGRAMMING  ***  \n\n\n");
   _outtext(buffer);
   fclose(fpr);}
   else  Verify();
abort1:  i=1;
}

/*************************************************************************/
/*                                                                       */
/*************************  VERIFY  DEVICE  ******************************/
/*                                                                       */
/*************************************************************************/
Verify()
{ long int i;
   int  j,k,color,ii;
   char  ansr;
   if  (choice=='C')
      {
      _clearscreen(_GCLEARSCREEN);
      M5:
      _settextcolor(11);
      sprintf(buffer,"ENTER  THE  NAME  OF  THE  FILE  TO  COMPARE  DEVICE\n");
      _outtext(buffer);
      _settextcolor(oldfgd);
      scanf("%s",name);
      if  ((fpr  =  fopen(name,"r"))==NULL)
         {_clearscreen(_GCLEARSCREEN);
         _settextcolor(12);
         sprintf(buffer,"%s  DOES  NOT  EXIST\n",name);
         _outtext(buffer);
         sprintf(buffer,"DO  YOU  WANT  TO  CONTINUE  ?   Y or N \n");
         _outtext(buffer);
         _settextcolor(oldfgd);
         ansr=toupper(getch());
         if  (ansr=='Y')goto  M5;
         else  goto  abort2;
         }

      line[0]='a';  line[1]='a';
      i=0;
         while(((line[0]!='*')||(line[1]!='L'))&&(i<20))
            {ReadLine(fpr);
            i++;}
      if(i>18){_clearscreen(_GCLEARSCREEN);
            _settextcolor(12);
            sprintf(buffer,"%s  IS  NOT  IN  STANDARD  FORMAT  !!!\n",name);
            _outtext(buffer);
            sprintf(buffer,"DO  YOU  WANT  TO  CONTINUE  ?   Y or N \n");
            _outtext(buffer);
            _settextcolor(oldfgd);
            ansr=toupper(getch());
            if  (ansr=='Y')goto  M5;
```

```
        else goto abort2;}
    i=0;
    while ((infile[i]=fgetc(fpr))!='*')
       { if (infile[i]=='\n');
         else i++;
       }
    }
    ii=0;
cont2:
if (( 8 & inp(inport))==0)
    {ii++;
    _clearscreen(_GCLEARSCREEN);
    _settextcolor(12);
    sprintf(buffer,"SUPPLY POWER IS OFF, PLEASE TURN IT ON");
    _outtext(buffer);
    for(i=0;i<ii;i++){sprintf(buffer,"!");
       _outtext(buffer);}
    sprintf(buffer,"\nDO YOU WANT TO CONTINUE ?  Y or N \n");
    _outtext(buffer); _settextcolor(oldfgd);
    ansr=toupper(getch());
    if (ansr=='Y')goto cont2;
    else goto abort2;}

    /* INIT DEVICE TO ID STATE */
    outp(outport,4); outp(outport,6); outp(outport,4); Execute();
    _clearscreen(_GCLEARSCREEN);
    for(k=1;k<3;k++)
    {ptr2=0;   eror=0;
    for (adrs=1;adrs<max+1;adrs++)
    {_clearscreen(_GCLEARSCREEN);
    _settextcolor(15);
    sprintf(buffer,"VERIFY IN PROGRESS %d\n",max+1-adrs);
    _outtext(buffer);
    if(eror==0)color=10;
       else color=12;
    _settextcolor(color);
    if(k==1)sprintf(buffer,"LOW ERROR = %d\n",eror);
       else sprintf(buffer,"HIGH ERROR = %d\n",eror);
    _outtext(buffer);
    Addr_gen();
    if(k==1)Ver_evenl();
       else Ver_evenh();
    Execute();
    Read_comp();
    ptr2++; /* increment file pointer for odd read */
    if(k==1)Ver_oddl();
       else Ver_oddh();
    Execute();
    Read_comp();
    ptr2++; /* increment file pointer for even read */
}
if(k==1)erlow=eror;
```

```
      else  erhigh=eror;
   }
      _clearscreen(_GCLEARSCREEN);  _settextcolor(15);
      sprintf(buffer,"*****  DONE  VERIFY  *****  \n\n");
      _outtext(buffer);
      if(erlow==0)color=10;
         else  color=12;
      _settextcolor(color);
      sprintf(buffer,"NUMBER  OF  VERIFY  LOW    ERROR  ========>  %ld\n",erlow);
      _outtext(buffer);
      if(erhigh==0)color=10;
         else  color=12;
      _settextcolor(color);
      sprintf(buffer,"NUMBER  OF  VERIFY  HIGH  ERROR  ========>  %ld  \n\n\n",erhigh);
      _outtext(buffer);
      fclose(fpr);
   abort2:  i=1;
}

/*********************************************************************/
/*                                                                   */
/*******************  READ  BACK  AND  SAVE  FILE  *******************/
/*                                                                   */
/*********************************************************************/
Rd_save()
{ M7:
   _clearscreen(_GCLEARSCREEN);
   _settextcolor(11);
   sprintf(buffer,"ENTER  OUTPUT  FILE  NAME  TO  SAVE  READ  BACK  DATA\n");
   _outtext(buffer);
   _settextcolor(oldfgd);
   scanf("%s",name);
   fpw=  fopen  (name,  "w");
      for  (adrs=1;adrs<max+1;adrs++)
      {_clearscreen(_GCLEARSCREEN);
      _settextcolor(15);
      sprintf(buffer,"READ  BACK  AND  SAVE  IN  PROGRESS  %d\n",max+1-adrs);
      _outtext(buffer);
      Addr_gen();
      Ver_even1();
      Execute();
      Read_store();
      Ver_odd1();
      Execute();
      Read_store();
      }
      _clearscreen(_GCLEARSCREEN);
      _settextcolor(15);
      sprintf(buffer,"********  END  OF  READ  BACK  AND  SAVE  ********\n\n");
      _outtext(buffer);
   sprintf(buffer,"FILE  SAVED  AS  ========>  %s\n\n\n",name);
   _outtext(buffer);
```

```
   _settextcolor(oldfgd);
   fclose(fpw);
}
/*********************************************************************/
/*                                                                   */
/*                            SHIFT/EXECUTE                          */
/*                                                                   */
/*********************************************************************/
Execute()
{outp(outport,5);  outp(outport,7);  outp(outport,5);  outp(outport,0);
}

/*********************************************************************/
/*                                                                   */
/*                  ADDRESS REGISTER SHIFT (00001)                   */
/*                  The rest of the instruction                      */
/*                  routines follow this format.                     */
/*                                                                   */
/*********************************************************************/
Addr_reg_sh()
{/* instruction data shifted LSB in first - (MSBxxxLSB) */
outp(outport,1);  outp(outport,3);  outp(outport,1);  outp(outport,0);
outp(outport,2);  outp(outport,0);  outp(outport,2);  outp(outport,0);
outp(outport,2);  outp(outport,0);  outp(outport,2);  outp(outport,0);
}

/*********************************************************************/
/*                                                                   */
/*       DATA REGISTER SHIFT (00010)            Data_reg_sh()       */
/*       GLOBAL BULK ERASE (00011)              Glob_bulk_er()      */
/*       PROGRAM EVEN COLUMNS (00111)           Prog_even()         */
/*       PROGRAM ODD COLUMNS (01000)            Prog_odd()          */
/*       PROGRAM SECURITY CELL (01001)          Prog_sec()          */
/*       VERIFY EVEN COLUMNS LOW (01010)        Ver_evenl()         */
/*       VERIFY EVEN COLUMNS HIGH (10010)       Ver_evenh()         */
/*       VERIFY ODD COLUMNS LOW (01011)         Ver_oddl()          */
/*       VERIFY ODD COLUMNS HIGH (10011)        Ver_oddh()          */
/*                                                                   */
/*********************************************************************/

/*********************************************************************/
/*                                                                   */
/*                    BULK ERASE PULSE WIDTH DELAY                   */
/*                                                                   */
/*********************************************************************/
BE_delay()
{ long int i,m,n;
   outp(outport,2);
   outp(outport,0);
m=z*5000;
  for  (i=0;i<m;i++)n=1;
}
```

```
/*********************************************************************/
/*                                                                   */
/*                    PROGRAM PULSE WIDTH DELAY                      */
/*                                                                   */
/*********************************************************************/
Prog_delay()
{ long int i,m,n;
   outp(outport,2);
   outp(outport,0);
   m=z*5000;
   for  (i=0;i<m;i++)n=1;
}
```

Section 1: Introduction to pLSI™ and ispLSI™

Section 2: pLSI and ispLSI Data Sheets

Section 3: GAL® Data Specifications

Section 4: pLSI and ispLSI Architecture

Section 5: pLSI and ispLSI Advantages

Introduction 5-1

Advantages of the Architecture 5-1

Advantages of E^2CMOS Technology 5-2

E^2CMOS Advantages Over Other Technologies 5-3

Advantages of in-system programming 5-4

Advantages of the pLSI and ispLSI Design Tools 5-6

Section 6: pLSI and ispLSI Software Development Tools

Section 7: pLSI and ispLSI Design Optimization

Section 8: pLSI and ispLSI Application Notes

Section 9: pLSI and ispLSI Article Reprints

Section 10: pLSI and ispLSI Quality, Reliability & Programmer Support

Section 11: Sales Information

pLSI and ispLSI Advantages

Introduction

Before the pLSI and ispLSI families were developed, groups of Design and Systems Engineers were interviewed to determine the true High Density needs of the engineering community. The engineers evaluated their current and future designs to determine the functions and system speed required, and the problems they were experiencing with other High Density devices. The answers were then evaluated and from the results the pLSI and ispLSI device families were developed.

Advantages of the Architecture

The pLSI and ispLSI families allow the designer to integrate multiple functions into one device and still maintain a high system speed and utilization. The families also give the designers a predictable delay, which allows them to predict the system speed before beginning a design or modifying an existing one.

The fundamental architecture of the pLSI and ispLSI devices is array-based. The devices offer a number of additional features which enhance functionality and simplify the design process. Architecturally the pLSI and ispLSI devices offer:

- Array-Based Architecture
  - Superior Performance
  - High Utilization
  - Predictable Delays
- Generic Logic Block (GLB)
  - Wide Function - 18 Inputs
  - Flexible Logic Path
    - High Speed (4 Product Term) Bypass
    - 20 Product Terms
    - Hardware XOR Functions
  - 4 Registered or Combinatorial Outputs
  - Asynchronous Control Product Terms
    - Clock
    - Reset to Registers
    - Output Enable
- Global Routing Pool
  - Predictable Fanout Delays
- Output Routing Pool
  - Flexible Routing to the Output Pins
- Advanced Clock Distribution Network
  - External and Internal Global Clock Generation

Array-Based Architecture

The first issues addressed in the pLSI and ispLSI families are the high system speed provided to meet or exceed today's fast clock speeds as well as the need for a predictable logic block and routing delays. For this reason, the pLSI and ispLSI devices were built with an array-based architecture which has numerous advantages over cell-based architectures, such as; superior performance, higher utilization of the internal logic and ease of design due to predictable delays.

There are two major factors that constrain a cell-based architecture. The first factor is associated with the interconnect structure. Cell-based devices use a channel routing structure similar to gate arrays. This type of structure gives the outputs of the logic blocks access to a limited set of routing resources. Hence, the output is a function of a limited number of signals. A major disadvantage of this routing scheme is the variable and unpredictable delays. On the other hand, pLSI devices use a centralized Global Routing Pool (GRP) which connects to the inputs of all the GLBs. Further, it allows every GLB access to all of the global signals, whether originating from the input pins, I/O pins or feedback from all GLBs. The advantages of this routing scheme are global connectivity, high product utilization and predictable and fixed delays.

GLB Features

The pLSI and ispLSI families were designed to incorporate both individual functions as well as complete systems. This flexibility is reflected in the wide inputs of the GLB. The wide inputs allow for easy design implementation of adders, decoders, counters, and state machines into a single level of logic, giving faster clock speeds and smaller delay times. At the same time, a small GLB with only four outputs also provides fast speeds for narrow input functions such as data manipulation, random logic or multiplexing, maintaining fast overall system speeds. Cell-based

devices, on the other hand, require multiple levels of logic to perform wide input functions which slows down the system speed.

There are several additional advanced features within the GLB including the Product Term Sharing Array (PTSA), high speed (4 PT) bypass and hardware XOR gates.

Product terms are broken up into four groups of 4, 4, 5 and 7. With the use of the PTSA, these product terms can be combined into virtually any combination, to provide up to 20 product terms per GLB output. The PTSA can route the terms to any of the 4 outputs of the GLB. Also, each of the 4 programmable outputs can be configured as a registered or combinatorial output. This gives the designer high utilization and flexibility of the GLBs.

When a design requires very high system speed, a configuration mode called 4 PT bypass is available to increase the speed through the GLB.

To increase the speed of a design with logic functions like parity generators, counters and adders, a hardware XOR gate has been built into every output path of the GLB. This enhances the speed of XOR based functions.

Asynchronous signals are also handled within each GLB. The designer has access to product terms which control the Clock, Register Reset and the Output Enable. This gives individual control of each GLB.

Global Routing Pool (GRP)

The GRP outputs can be connected to all the global signals, whether originating from the I/O pins or as feedback from the GLBs. The advantage of this routing structure is its global connectivity and its high performance fixed and predictable routing delays. Hence, it is easy to predict the system speed prior to designing with the pLSI and ispLSI families. The dedicated inputs, which do not run though the GRP, also have a fixed routing delay.

Output Routing Pool (ORP)

GLBs are not locked to one output pin, instead there is an ORP that gives flexibility in output pin assignment. The ORP allows the design to be implemented with little regard for logic placement. This helps the designer achieve high product utilization and maintain pinouts for incremental design changes.

Advanced Clock Distribution Network

The pLSI and ispLSI families have a complete Clock Distribution Network. The Clock Distribution Network allows the designer to construct the system clocks both externally and/or internally through a special "clock GLB". In addition to its standard connections, the "clock GLB" also has a direct connection to the Clock Distribution Network. Asynchronous, logic function and "divide by" clocks can be easily designed within the pLSI and ispLSI devices and directly connected to the global clock nets. The global clock nets give the design a smaller skew between all of the GLBs compared to product term clocks.

Each I/O pin has an I/O cell associated with it. The I/O cell can be configured for different functions such as input registers or input latches. The correct ratio of I/O pins to the internal logic is essential for efficient design implementation. The pLSI and ispLSI family has an I/O ratio of 2 I/O pins for every GLB allowing most designs to fit easily. Presently, the family ranges from 44-pin to 120-pin devices, which accommodates a wide range of design requirements.

Advantages of E^{2}CMOS Technology

The pLSI and ispLSI devices are manufactured using Lattice's proprietary high speed 0.8 Micron UltraMOS IV E^{2}CMOS process. This process successfully combines the best features of CMOS and NMOS process technologies. Devices made with this process offer:

- High System Speeds
- Very High Logic Densities
- Low Power Consumption
- Non-Volatile, in-system programmability
- Fast Erase and Reprogram Times
- 100% Tested
- Guaranteed 100% Programming and Functional Yields
- Lower Manufacturing and Support Costs

Lattice's UltraMOS IV E^{2}CMOS devices offer an ideal combination of high speed, high logic density and low power. They offer the highest system speeds of any high density device on the market today, and do so at extremely low power consumption levels.

The reprogramming and in-system programming features ensure significant cost savings in the engineering laboratory, on the production floor and in the field. Because the parts are 100% tested at the factory, both manufacturing and field support costs are greatly reduced. These features combine to make the pLSI and ispLSI devices the best choice available in high density devices.

Testability

The most significant advantage of E²CMOS over competing technologies is its inherent testability. Capitalizing on very fast (200 ms) erase times, Lattice repeatedly patterns and erases all devices during manufacture. Lattice tests each pLSI and ispLSI device for AC, DC and functional characteristics. The result is guaranteed 100% programming and functional yield.

Low Power

One of the many advantages of E²CMOS technology is it's low power consumption. CMOS provides the immediate benefit of decreased system power requirements allowing for high reliability and cooler running systems. CMOS technology also permits circuit designs of much higher functional density, due to lower junction temperatures and power requirements on chip. The user benefits, because higher functional density means further reduction of chip count and smaller boards in the system.

High Speed

Also advantageous is the very high speed attainable with Lattice's state-of-the-art E²CMOS process. Lattice pLSI and ispLSI devices are faster than any other currently available solution.

Prototyping and Error Recovery

Finally, E²CMOS gives the user instant erasability with no additional handling or special packages necessary. E²CMOS products are ideal for prototyping because designs can be revised instantly, with no waste and no waiting. On the manufacturing floor instant erasability can also be a big advantage for dealing with pattern changes or error recovery. If a pLSI or ispLSI device is accidentally programmed to the wrong pattern, the device can be easily reprogrammed. No other technology offers this advantage.

E²CMOS Advantages Over Other Technologies

Programmable Logic Devices are generally manufactured using one of five fuse technologies:

- Bipolar Fuse-Link
- Anti-Fuse
- UV-EPROM Cells
- SRAM Cells
- E²CMOS Cells.

A comparison of E²CMOS memory cells with other programmable technologies is shown in table 5-1.

Table 5-1. Technology Comparison

Cell Technology	Lattice E²CMOS	Bipolar Fuses	Anti-fuse	UV-EPROM	SRAM
Erasable and Reprogrammable	**Yes**	No	No	Yes	Yes
Programming Time	**Fast**	Fast	Slow	Slow	Fast
Erase Time	**Fast**	N/A	N/A	Slow	N/A
Testability	**Excellent**	Poor	Poor	Good	Excellent
Reconfigurable in-system	**Yes**	No	No	No	Yes
Non-volatile	**Yes**	Yes	Yes	Yes	No
Other Comments	**in-system programmable**	High Power Consumption	–	Expensive Windowed Package	External Circuitry Required

Bipolar Fuse-Link Technology

Bipolar PLDs are typically fast, but consume a lot of power. E²CMOS devices achieve speeds similar to bipolar devices, but consume less power and run at cooler temperatures. Because less heat is generated, the E²CMOS parts are capable of much higher logic densities.

Another limitation of the Bipolar technology is the one-time-programmable fuses. Complete testing of bipolar devices is impossible because the fuse array cannot be tested before programming. Bipolar device manufacturers rely on complex schemes using test rows and columns to simulate and correlate their device's performance. The result is programming failures at the customer site. Any misprogrammed device due to mistakes during prototyping or errors on the production floor must be discarded because bipolar devices cannot be reprogrammed.

Anti-Fuse Technology

Devices using anti-fuse programming technology have restrictions very similar to bipolar technology. The anti-fuses can only be programmed once, and therefore suffer the same reusability and testability limitations as bipolar fuses. In addition, the programming times can be very long, slowing down both the development cycle and the production process.

UV-EPROM Technology

E²CMOS devices also offer advantages over UV-EPROM cell based devices. These devices must be removed from the target system to be erased and reprogrammed, while E²CMOS devices can be conveniently erased and reprogrammed in-system. This eliminates the need for sockets and is especially attractive when surface mount devices are used.

UV-EPROM devices require approximately 20 to 30 minutes to erase, while E²CMOS devices are erased in less than 200 milliseconds. Because the UV-EPROM device erase time is long, manufacturers typically limit the testing of these devices.

Additionally, the devices must be housed in expensive windowed packages to allow users to erase them. As a cost-reduction measure, UV-EPROM device manufacturers also offer their devices in windowless packages. Although windowless packages are less expensive, they cannot be completely tested or reprogrammed. The windowless packages eliminate the reprogramming option available in the UV-EPROM technology.

SRAM Technology

Some High Density devices use SRAM cells to store their programming data. Like all SRAMs, these parts lose their memory when the power is removed, and must be reprogrammed each time the system is powered up. This can be done by either adding an extra memory chip to store the configuration data, or by configuring the part from the system processor. With either method, there is a delay associated in configuring the device. The E²CMOS devices are non-volatile and eliminate this delay.

E²CMOS Technology

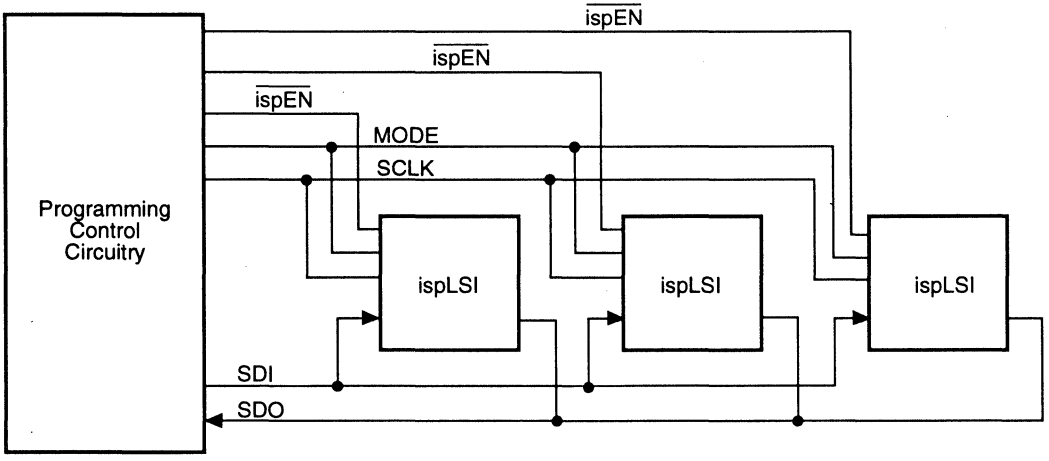
Testing is a major issue to customers when using devices of this complexity. The reprogramming of the pLSI and ispLSI devices offers many testing advantages to the user. Each device is tested at the factory for functionality, programmability and speed. This guarantees that no programming or functional failures will occur.

The reprogramming feature of the device can be used as a system testing feature. During the manufacturing process, the device can be loaded with test programs for testing circuitry around the pLSI or ispLSI device. When that circuitry has been checked out, the device can then be reprogrammed in its final configuration. This can be done either on the test bench or in a bed of nails type tester. No other programmable technology offers these advantages.

Advantages of in-system programming

When programming a device, regardless of the technology, high programming voltage and high programming current must be created to place the logic connection (or disconnect the logic) in a non-volatile programmable memory cell or a programmable fuse. Having a programmable logic device in a design gives the designer the ability to make design changes late in the design cycle and not have to implement a rework on the board. An ideal process for all the design changes would be to reprogram the devices on the board itself or in-system. Of the currently available high density programmable logic devices on the market, only static memory cell based devices have this potential of reprogramming without removing the devices from the board. One of the major drawbacks of the static memory cell based devices is that the volatile memory must be reconfigured every time the power is turned off. The ispLSI devices enable design changes to be implemented on the board without removing the device from the board. By using E²CMOS technology the user can turn off the power and not lose the logic configuration. Lattice

Figure 5-1. in-system programming Interface



guarantees the data retention in excess of 20 years and 1,000 programming cycles.

The ispLSI devices are 100% compatible with the pLSI versions. By integrating the programming circuitry on chip, Lattice ispLSI devices offer the best features of both non-volatile programmable logic devices and the ability to in-system program the devices. The on-chip programming circuit provides all the necessary programming voltage and current levels from the standard 5V power supply. The programming is based on the five programming interface signals, isp Enable ($\overline{\text{ispEN}}$), Serial Data In (SDI), Serial Data Out (SDO), Serial Clock (SCLK) and Mode control (MODE). Figure 5-1 is a block diagram of multiple ispLSI devices and how the five interface signals are connected. This enables the designer to do prototyping, development, and remote access programming with the ispLSI version. For a detailed explanation on how ispLSI devices work, refer to section 4.

From the designer's point of view, the isp feature simplifies design changes, manufacturing flow, increases reliability, and shortens the design cycle. With the on-chip programming circuitry, the benefits of in-system programming vastly outweigh the small overhead incurred by the implementation. The implementation is as simple as providing the five programming signals (in accordance with the programming requirements) and down loading the programming file to the device.

Benefits of in-system programming

Of the many benefits of in-system programming, some are quite subtle. But these subtle benefits could very well revolutionize the way the systems are designed, manufactured and serviced. A brief description of each of the following four areas, indicates how these benefits can be used to improve designs in different applications:

- Board Design and Manufacturing Flexibility
- System Diagnostics
- Software Reconfigurable Systems
- Easier Field Updates

Board Design and Manufacturing Flexibility

The ability to custom design and modify the logic on chip with the high density devices, allows the designers to define the board layout earlier in the design cycle which in turn shortens the design cycle. The Lattice ispLSI devices make this process easier by eliminating the need for a separate programmer. Since the reprogramming is done in-system, there is no need to remove the device, thus eliminating the requirement to have a socket for the device. The removal of sockets improves reliability and reduces the cost of the system. In manufacturing, removal

of the separate programming sequence on the programmer streamlines the manufacturing process and reduces the chance of introducing programming-related errors. If desired, the ispLSI devices can still be programmed with a device programmer just like the pLSI devices.

To better illustrate this design and manufacturing flexibility, let us take an example where an unforeseen event in the system forces the designer to make a design change after the design has been completed. If the design change only involves logic modifications to the ispLSI devices that are soldered on the board, the design change is as simple as generating the new programming files for the devices and taking the boards back through the in-system programming process. The process would be much more complicated, expensive and time consuming if the devices were not ispLSI devices and had to be removed and reprogrammed.

System Diagnostics

With the ispLSI devices, system diagnostics can be designed into the system. By placing a test pattern into the ispLSI device when the system is in test or diagnostic mode, the functional logic can be bypassed to test specific portions of the board design. After the test is done, the functional pattern can be reprogrammed into the device for normal operation.

The internal registers of the ispLSI devices can be read or written to, by using the register preload feature. Reading the registers allows the system to diagnose system failures based on the internal states of the registers. Being able to write to the internal registers allows the designer to force the system, for diagnostic purposes, into states that test the logic.

Software Reconfigurable Systems

The options are limitless to the designer who has the ability to change the functionality of a device that is soldered on a board. Different configurations of the hardware can be implemented on the same board when the configuration logic is programmed into the ispLSI devices. A system can be designed so that it is adaptable to the external environment. The different features that are supported by different versions of the hardware can now be implemented with one version of the hardware which is changeable via software control. Modifications of the board addresses or hardware configurations for different system interfaces can now be done with software. Optional features can now be added to the hardware simply by reprogramming the ispLSI devices with the new pattern.

Easier Field Updates

With software reconfigurable systems, updates are as easy as loading the new configurations from a floppy disk or downloading them via a modem. Often this may be the only way to do an enhancement on a system that is located in a remote area or in a location that is a health hazard. The direct result of this feature is a quicker update or service time for customers. Not having to build a completely different board for the additional features will reduce the cost of supporting the additional features. Diagnostic features can be implemented so that some field diagnostics are controlled and monitored by the software from a remote location.

Advantages of pLSI and ispLSI Design tools

Since many manufacturers offer high density devices, the development software for these products can influence the selection of one manufacturer over another. The pLSI Development System (pDS) is an entry level system which runs on an IBM PC. Designs are entered into the pDS software using Boolean Equations and Macros. It offers many advantages over other high density design packages:

- Low Cost
- Graphical User Interface
- Flexible Design Entry
- Large Number of Macros
- Quick Compilation

Inexpensive

Designing with the pDS software is inexpensive for several reasons. It is one of the least expensive design packages on the market today. The pDS software runs on most IBM compatible 286 and 386 based PCs, reducing the cost of expensive high speed computers or workstations. The software is also easy to learn and use, substantially reducing design cycle time and saving valuable engineering resources. In addition, using in-system programming eliminates the need to buy an expensive software simulator or device programmer.

Graphical User Interface

The pDS software runs under Microsoft Windows 3.0 on an IBM PC/AT or equivalent. Using the Windows interface, the pDS software displays a graphical representation of the block diagram of the pLSI or ispLSI device.

The program makes extensive use of pull-down menus, and is very easy to learn. All commands are entered using the mouse, including the design entry, edit, cut, and paste commands.

Since the pDS software runs under Windows, compatibility with different computers is automatic. Microsoft Windows runs on almost all IBM PC/AT compatible computers. Windows drivers are available for most of the graphics cards and printers sold today. If a computer runs Windows in the standard or 386 enhanced mode, it will run the pDS software.

Flexible Design Entry

In the pDS software, designs are entered into the pLSI or ispLSI devices using a combination of Boolean Equations and Macros. Since the syntax is similar to the ABEL program used for developing GAL devices, the time that it takes to learn the software is substantially reduced. Existing GAL designs can be transferred into pLSI or ispLSI devices by "cutting" the equations from ABEL files, and "pasting" them into the Lattice pDS software.

To enter a design, the designer simply clicks on the GLB or I/O Cell and enters the desired Boolean Equations or Macro for that cell through the edit window. Standard Windows *"Cut* and *Paste"* commands reduce redundant typing, and Macros make entering large blocks of logic fast and simple. As the logic for each block is entered, it can be locally verified to ensure that there are no syntax or logic errors. Finding errors early in the design cycle prevents them from being compounded and repeated.

The pDS software supports simulation of pLSI and ispLSI devices by exporting interface files compatible with Viewlogic's Viewsim simulator. Completed designs are simulated by importing the simulation file into Viewsim.

Large Number of Macros

The Lattice pLSI/ispLSI Development System comes with a comprehensive library of Macros. Macros duplicate many of the same logic functions found in the 7400 series functions. These functions range from simple gates and registers to complex counters, multiplexers and adders. To design with Macros, the designer simply selects them from the library and then interconnects them as needed. Macros are divided into two groups: Hard Macros and Soft Macros. Hard Macros are Macros which have been optimized to fit into the pLSI and ispLSI architecture and cannot be modified. Soft Macros are general purpose logic blocks that can be modified. If a function is needed that is similar to one provided in the standard library, it can be modified to create a new Macro. The new Macro is then saved to a User-Macro library. Over 90% of the Macros in the standard library are Soft Macros.

User-Macro library elements are created as needed. These Macros are a combination of Boolean Equations and existing Macros. Once a new Macro has been created, it can be stored in a User-Macro library. Of course, as the User-Macro library grows, designs are created faster and more easily.

Quick Compilation

Once the design has been entered, it is then compiled. The compilation cycle consists of three steps. The first step is design Verification which checks for such errors as syntax, missing or duplicated networks and logic errors. The next step is Routing which interconnects the GLBs and I/O Cells according to the designer's specifications. The last step is the generation of a programming file which conforms to the JEDEC standards. This programming file is similar to the one used for programming GAL devices.

Running these steps is simple and fast. The GRP and ORP are designed to decrease the routing time from several hours to less than a few minutes. Using the pDS software, all three steps take approximately ten minutes on most PCs. If corrections are needed to the original design, they can be done quickly. Fast design cycle time results in both greater engineering productivity and reduced time to market.

The in-system programming eliminates time-consuming simulation because the design is downloaded and tested directly to the target system. If errors are found, they are corrected and a new programming file is quickly generated.

The Lattice pLSI/ispLSI Development System is designed to be simple enough to be used by any novice designer, and comprehensive enough to be used by the most exacting of designers. Low cost, compatibility with existing PCs and fast design cycle times should make it attractive to all designers.

Section 1: Introduction to pLSI™ and ispLSI™

Section 2: pLSI and ispLSI Data Sheet

Section 3: GAL® Data Specifications

Section 4: pLSI and ispLSI Architecture

Section 5: pLSI and ispLSI Advantages

Section 6: pLSI and ispLSI Software Development Tools

System Design Process 6-1

pLSI and ispLSI Design Flow 6-3

pLSI and ispLSI Development System 6-5

The User Interface 6-5

Entering a Design 6-8

Verifying a Design 6-10

Lattice Place and Route 6-11

Generating the Programming File 6-13

Download to a Device 6-14

Report Files 6-14

Clock Distribution Network 6-20

Macro Usage 6-22

Logic Simulation of the Design 6-27

Section 7: pLSI and ispLSI Design Optimization

Section 8: pLSI and ispLSI Application Notes

Section 9: pLSI and ispLSI Article Reprints

Section 10: pLSI and ispLSI Quality, Reliability & Programmer Support

Section 11: Sales Information

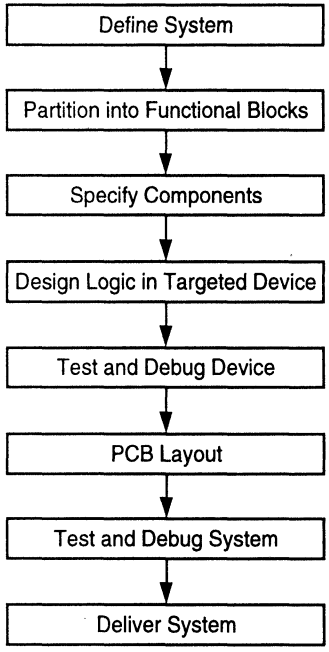
System Design Process

Introduction

Conceptually defining a design is the first step in the design process. This involves visualizing the design's interaction with the rest of the world and defining a general flow diagram to determine the design's basic sequential behavior. This organizational flow, used to integrate the design subsystem into high density devices, is described in the following topics and is shown in figure 6-1.

- Partitioning
- Specifying Components
- Logic Design and Optimization
- Test and Debug
- Printed Circuit Board Layout
- System Test and Debug

Figure 6-1. System Design Flow



Partitioning

After completing the system conceptually, the designer partitions the system into modules or functional blocks. These blocks can be a few components or multiple circuit boards with numerous components. The designer organizes these functional blocks to meet the capabilities of the devices being targeted, for example, the number of I/O pins, flip-flops and gates needed. The user should also consider the frequency at which the targeted device must operate, the number of clocks required, and the timing relationships of signals (AC specifications).

Specifying Components

After the partitioning is defined, the designer chooses which components will be used to accomplish the task. The design should meet the system specifications using the least number of components required to keep the system cost as low as possible, while keeping the system reliability as high as possible.

With system specifications often calling for low weight, low power and reduced size, designers require higher levels of integration. If the design tools and solutions are not available, these added requirements affect the design schedule and project completion. The pLSI and ispLSI high density devices can solve many such design requirements. The pLSI and ispLSI family of high-speed, high-density pLDs also include easy-to-use software for design and verification.

Logic Design and Optimization

After the functional partitioning and component specifications are completed, the logic necessary to implement the functions is defined. The logic includes standard TTL functions, CMOS logic functions, or functions from a library, such as the Lattice Macro Library. The implementation of logic into a high density device is optimized for the targeted device. The partitioning also effects the optimization. Optimization can be for speed, utilization or a combination of both.

Logic entry for a Lattice high density device is done with the pLSI/ispLSI Development System (pDS). The pDS software operates under Microsoft Windows 3.0. With the pDS software, the designer can go through the complete design flow, from logic entry to programming pLSI and ispLSI devices within hours.

Test and Debug

When designing a system, or a portion of a system, it is easier to test and debug pieces or modules rather than the entire system. In this manner, the designer can confirm modules, or functional blocks, and find problems earlier in the design cycle.

Logic can be tested by either simulation, or actual testing of the programmed device. Simulation can be accomplished using the Viewsim logic simulator. Design errors detected by software simulation are corrected by the designer, before the printed circuit board is laid out and manufactured, which saves time and reduces cost.

Reprogrammability allows the designer to test, debug, and modify logic without a simulator. The pLSI and ispLSI devices can be reprogrammed multiple times. This reprogrammability further assists the designers by allowing them to program the device with diagnostic and verification logic.

The designer should always attempt to design the logic with testability in mind. Testability means different things to different designers. Key guidelines to be aware of are:

- Large Counters Should be Segmented for Easy and Fast Testing.
- Logic Should be Designed for Controllability and Observability.
- There Should be no Floating Nets.
- All Nets Should be at a Known State or Able to be Set or Reset.

The ispLSI devices offer preload and verification features. These features allow register contents to be verified without using logic analyzers or other debugging tools. For details, refer to ispLSI Advantages in Section 5.

Printed Circuit Board Layout

Once the logic has been verified, the Printed Circuit Board (PCB) is laid out and manufactured. Since the logic may be changed, this phase of the system design is usually executed after the logic has been validated. It is recommended that the board design and layout be done after designing the pLSI and ispLSI parts.

System Test and Debug

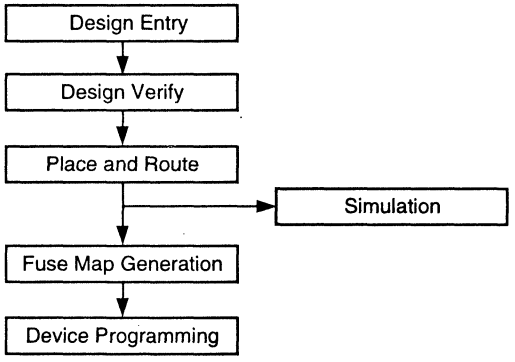
This is the final stage of the design process. The logic and the PCB are tested as a system and minor enhancements or bug fixes are implemented. Because of the flexibility of the pLSI and ispLSI devices, minor changes can be made without affecting the layout of the PCB and even the pinout of the device.

pLSI and ispLSI Design Flow

Introduction

Once the system design has been broken down into components, and the logic functions which need to be incorporated in the selected components defined, the logic design phase begins. The general design flow is shown in figure 6-2. Lattice offers the pLSI/ispLSI Development System (pDS) for designs being implemented into a pLSI or ispLSI device.

Figure 6-2. General Design Flow



The pDS software is used to implement the design from logic entry through programming the device. The design process using the pDS software is a basic six step process as shown in figure 6-3 and further described in this section.

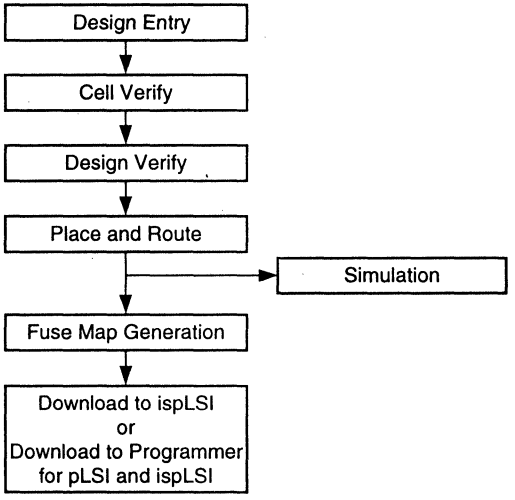
Design Entry

With the pDS software, the user partitions the logic manually. This is done by grouping logic equations so that they fit into the resources available in Generic Logic Blocks (GLB) and I/O Cells. The designer enters the logic equations in Boolean format through a Windows based environment.

Cell Verification

After the logic has been entered into a GLB or I/O Cell, it is verified. The Verify function ensures that the logic will fit, minimizes the logic if appropriate, and verifies that all the logic is connected by signal names. The user has optional directives that can override certain actions, such as minimization.

Figure 6-3. pDS software Design Entry



Design Verification

After all the logic has been entered, a design verification is completed. This verification first checks any cells which were not verified previously with the Cell Verify and then checks all nets for proper termination.

Place and Route

After a successful verification, the pDS software automatically enables the Lattice Place and Route (LPR) menu item. The LPR maps the entered logic to the architecture of the selected device. The user may lock or fix signals to specific pins or let the LPR automatically select the pins.

Once the LPR is complete, the design can be exported in an Electronic Data Interface Format (EDIF) or a network file. These options allow the designer to communicate with other design environments or perform simulation.

FuseMap

The final step before programming the device is to generate the fuse file. The *FuseMap* Command reads the LPR database and generates a fuse file. The fuse file is used to program the selected device. It is an ASCII file and is in the JEDEC format. This operation has one option. The security cell can be either enabled or disabled.

Device Programming

Two programming methods are used to program pLSI and ispLSI devices. The first method uses the Device Programming Mode for both ispLSI and pLSI devices. This method uses device programmers from third party vendors. The second method uses the Lattice in-system programming Mode and applies to the ispLSI family of devices.

Both methods of device programming allow the user to program and read back the fusemap from the programmed device for verification (if the security cell has not been set).

pLSI/ispLSI Development System (pDS)

Introduction

The Lattice pLSI/ispLSI Development System (pDS) is a software package used to implement designs in pLSI and ispLSI devices. It allows users to enter and verify individual I/O cells and GLBs using a simple graphical interface. The following pDS software topics are discussed in more detail in this section:

- System Requirements
- The User Interface
- Entering a Design
- Verifying a Design
- Using the LPR
- Generating the Programming File
- Download to a Device
- Report Files
- Clock Distribution Network
- Macro Usage

System Requirements

The pDS software can be run on 286, 386 and 486 IBM PC compatibles. The recommended system configuration is:

- IBM PC 386 or greater
- VGA or Higher Resolution for the Display
- 4 Megabytes of RAM
- 60 Megabyte Hard Disk
- 1 Serial Port
- 1 Parallel Port
- Mouse
- DOS 3.3 or greater
- Microsoft Windows 3.0

The User Interface

The Graphical User Interface (GUI) is the display which the designer sees when they invoke the Lattice pDS software (see figure 6-4). The diagram or picture that is presented is a representation of the device with which the designer is going to work. With the GUI, the designer can understand the layout of the device and the different sections, such as the GLBs, and the I/Os. The designer also uses the GUI to enter the logic and perform the necessary functions to program a device.

The Lattice pDS software incorporates a menu system which operates similar to other Microsoft Windows applications. The GUI includes the menus, and the menu items needed to complete an entire design. For a complete description of all the menus and all their options, please refer to the pLSI/ispLSI Development System User Manual.

The Menu Bar along the top lists the menus available. If a menu or a menu item is grayed out , this means that the menu or the menu item is unavailable to the designer at this time. Some menus and menu items will be enabled when the prerequisite steps have been completed. This will ensure all the design steps are followed in the correct sequence.

The Lattice pDS Software has nine main menus to choose from:

- *File*
- *Design*
- *Cell*
- *Macro*
- *Library*
- *Zoom*
- *Search*
- *Message*
- *Help*

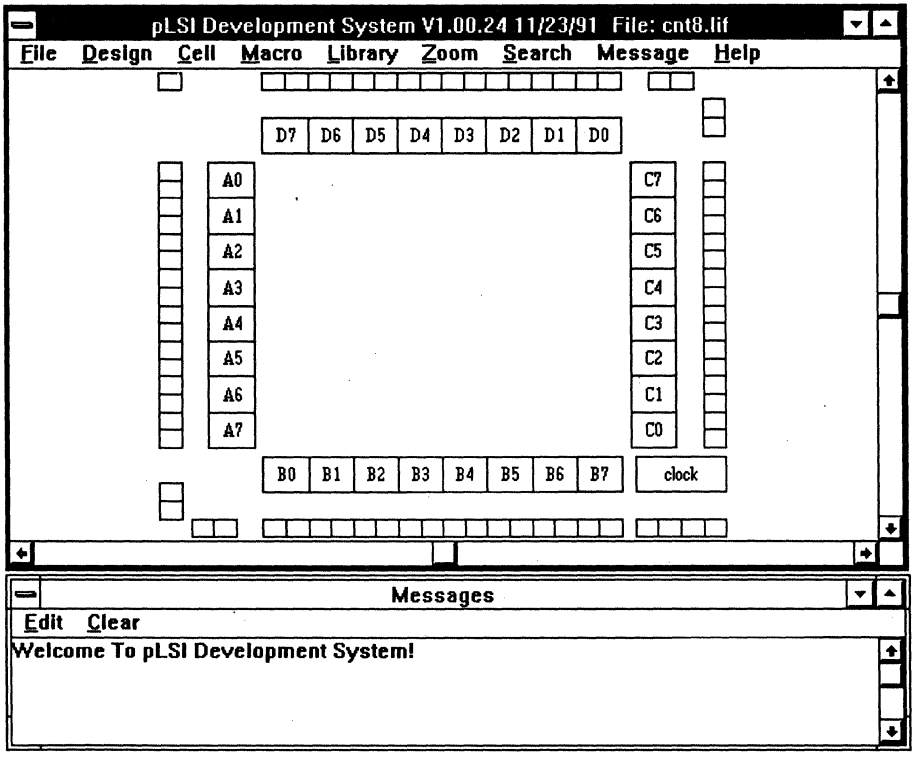
File Menu

The first menu, *File*, includes several items as shown in figure 6-5. Some of the items are submenus. The *File* menu allows users to save, read or create new design files. It generates the design status report files and allows users to print a design file as well as reports.

The *New* menu item clears the design library and brings the display back to the same as it was when the software was first started (default size). It establishes a design environment to enable users to create a new design.

The *Open* menu item opens a design that was previously created and saved. The designer can open a design that is in the .lif format. This format is the Lattice Internal Format which is a binary file, and is not readable, except by the pDS software.

Figure 6-4 Graphical User Interface



The *Save* menu item saves the design. The design may be assigned any name up to eight characters or numbers using standard DOS naming conventions.

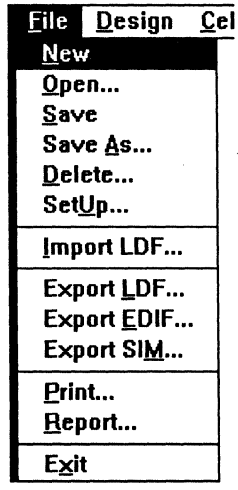
The *Save As* menu item saves a design to a file with a different name. Designers can also specify where the file is to be located, such as a different directory and/or drive. This may be helpful when working on multiple projects or when more than one person is using the computer. This command is also used to name an "unnamed" design.

The *Delete* menu item is used to delete the current file.

The *SetUp* menu item has multiple functions. It allows users to:

- Select a Device
- Define the Library Search Order and Add Libraries
- Enter Comment Information such as Design Name, Revision, Author and Project
- Enter a Description
- Change Default Minimization Setting

Figure 6-5. File Menu



Except for the device selection and design name, all of the information in the setup menu is optional.

The *Import LDF* menu item reads in the design_name.LDF (which was either created outside pDS software environment using a text editor or previously exported from pDS software) and translates it into the Lattice Internal Format (LIF).

The *Export LDF* menu item translates the design and saves it as design_name.LDF, which is an ASCII or text format file. This file retains all user defined logic as well as the name of system library functions. This file can be imported back into the Lattice Design System.

The *Export EDIF* menu item is available to translate from the .LIF format to the EDIF netlist format. This format can be read by other systems.

The *Export SIM* menu item enables the designer to translate the design into a Lattice Simulation Netlist. This format has an extension of .SIM. The *Export SIM* command is selected if the designer wishes to simulate or verify the logic in the design.

The *Print* menu item prints any text that has been selected and put into the clipboard (copy or cut) or prints a report file. Refer to the report file portion of this section for details about these files.

The *Report* menu item enables the designer to create report files. By selecting the report item, a report generation box will appear with the report files available.

The *Exit* menu item closes the Lattice pDS software and returns the designer to the Program Manager of Windows.

Design Menu

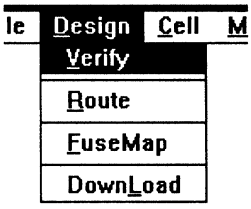
The second menu is the *Design* menu. The design menu allows users to go through the basic steps of the pLSI & ispLSI design process (refer to figure 6-6). The design menu has four items:

- *Verify*
- *Route*
- *FuseMap*
- *DownLoad*

Verify checks to see that all signals have a source and a destination. It performs a Cell Verify on GLBs and I/O cells that have not been verified.

Route connects the logic as specified in the design. Route also assigns pins if they were not previously fixed.

Figure 6-6. Design Menu



FuseMap generates a JEDEC file with the information from the Lattice Place and Route (LPR) database.

DownLoad communicates with either a device programmer or directly with an ispLSI device. Download transfers the JEDEC programming file to the device directly or to a device programmer.

Cell Menu

The *Cell* menu allows the designer to *Edit*, *Copy*, *Move*, *Delete*, and Name a GLB or an I/O cell. These options are similar to standard Windows operations, except for *Name* which replaces the default label of the GLB or I/O cells for easier identification.

Macro Menu

The *Macro* menu opens a listing of available Macros from the currently selected library. User-defined Macros can also be created. Refer to the Macro Usage discussion later in this section to get an explanation of the use of Macros and how to create them.

Library Menu

From the *Library* menu, the designer can select a current library or delete a library. Refer to the Macro Usage later in this section.

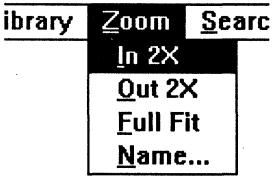
Zoom Menu

The *Zoom* menu allows the designer to resize the image on the display (Refer to figure 6-7). One of the most frequently used options in this menu is the *Full Fit*. If the user resizes the window this option will size the image to the full size of the window. It can be used at any time. The other zoom features are discussed in the pDS Software User Manual.

Search Menu

The *Search* menu provides options for performing global or selective searches. Either type of search eliminates the need to scroll through lists in any window to locate specific information.

Figure 6-7. Zoom Menu



Message Menu

The *Message* menu opens the message window if it is closed. The default for the message window is open and it is below the main window when the pDS software is invoked.

Help Menu

The *Help* menu is an on-line help feature, similar to the help in other MS Windows applications.

Entering a Design

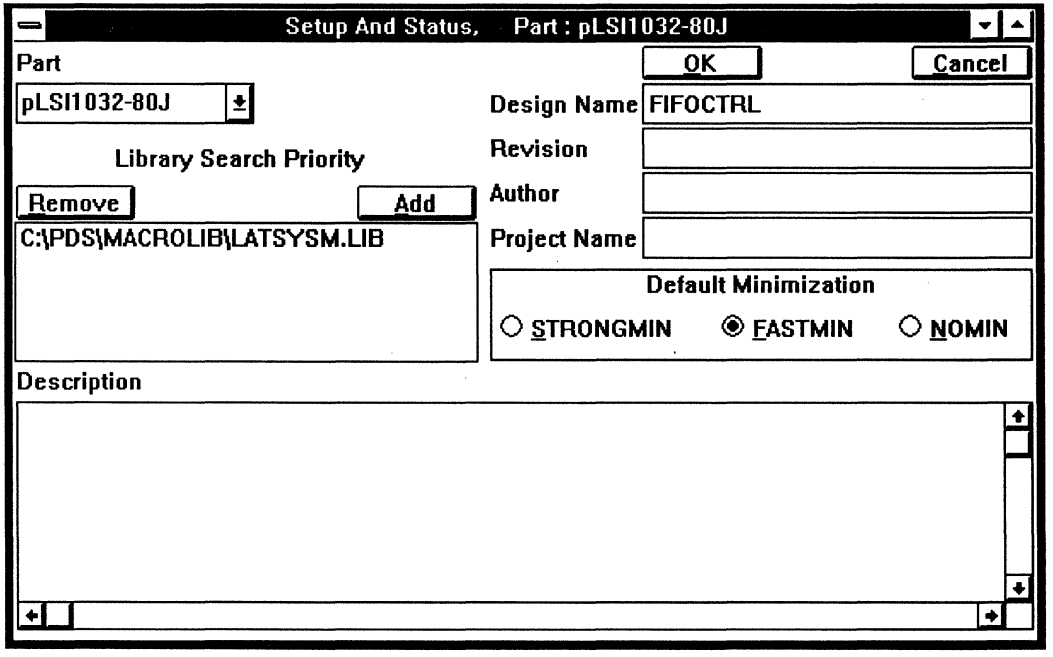
Entering a design using the pDS software is a straightforward task. The designer enters logic using Boolean equations or Macros which emulate SSI/MSI TTL type functions. This section discusses the properties of entering a design, including:

- Equations
- Macros
- Syntax (signal order and keywords)

Before entering a design the following steps must be performed:

1. Select the menu item *SetUp* from the *File* menu. The Setup and Status window is invoked as shown in figure 6-8.
2. From the Setup and Status window, select *Part*. For the software to be configured correctly, this information is required.

Figure 6-8. Setup and Status



Some of the information from the device file is also used for Verify, LPR, and FuseMap.

For a step by step tutorial of a simple design, from entry to programming, refer to the Beginner's Guide in section 8.

At this point it is assumed that the design has been partitioned to fit into the GLBs, I/Os and the design itself will fit into the targeted device.

The design may be entered in a GLB in either equations or Macros or a combination of both. While the logic in an I/O cell may only be in Macro format since all input, output and bidirectional I/O functions are hard Macros.

When entering logic in a GLB, and using Macros or Boolean equations, the Macros must be entered before the Equation section. The keyword EQUATIONS, defines the beginning of equations and the keyword END defines the end of the equation section as shown in example 6-1.

Example 6-1. Boolean equation Format

```
Equations
    CO = q0 & q1 & q2 & q3;
end
```

where:

CO is the signal name that is being defined and is on the left followed by an = sign.

q0, etc. forms the Boolean equation which defines the function of CO.

The Boolean (logic) operators used in the pDS software are similar to ABEL™ operators and equations with one exception, the $$. The $$ operator tells the software to use the internal Exclusive-Or (XOR) gate that is in the GLB rather than an AND/OR equivalent which is less efficient. All the logic operators are listed in table 6-1. The precedence in table 6-1 refers to the order in which the software evaluates an equation. Using parentheses, overrides this precedence.

Table 6-1. Boolean Logic Operators

Operator	Precedence	Definition	Example
!	1	NOT	!A
$$	2	XOR (Internal)	A $$ B
&	3	AND	A & B
#	4	OR	A # B
$	5	XOR	A $ B

As in ABEL, the designer can also use *Dot* extensions. Dot extensions are used to describe logic functions such as the *D, Clock, Reset, Q* of flip-flops or output enable (.OE). The dot extensions available within the software are listed in table 6-2 .

Table 6-2. Dot Extensions

Dot Extension	DESCRIPTION	Left of =[1]	Right of =[3]	Macro Usage[3]
.D	Data input of a D flip-flop	X	X	
.Q	Register feedback[4]		X	X
.PIN	Combinatorial logic feedback		X	X
.RE	Reset of a register	X		
.CLK	Global clock for a register	X		
.PTCLK	Product Term clock for a register	X		X[5]
.OE	Output Enable	X		

1. The dot extension can be used on the left side of an equation.
2. The dot extension can be used on the right side of an equation.
3. The dot extension can be used to define the input/output signals for a macro being used in a GLB.
4. Implied for a registered signal that appears on the right side of an equation.
5. Can be used on the clock input for an async macro to force product term clock usage rather that the defined system clock (see figure 6-20).

Example 6-2, 6-3 and 6-4 illustrate the use of dot extensions. Example 6-2 shows a T-type flip-flop implemented using the combinatorial XOR function and clocked with a product term clock.

Example 6-2.

```
Equations
   Tout.d = Tout.Q $ Tin;
   Tout.ptclk = sig1 & sig2;
End
```

Example 6-3 shows a T-type flip-flop with a synchronous or global clock and product term reset using the combinatorial XOR function.

Example 6-3.

```
Equations
   TinX1 = A # B # C # D # E # F;
   Tout.d = Tin.Q $ TinX1;
   Tout.clk = XCLK1;
   Tout.re = G & H;
End
```

Example 6-4 shows how an Output Enable (OE) is defined using the .OE extension.

Example 6-4.

```
Equations
   WR.oe = CS & !Rd;
End
```

Lattice software is not case sensitive, i.e. upper case and lower case letters are treated the same.

Verifying a Design

Design verification is done after the designer has entered the design and it consists of two processes:

- Cell Verification
- Design Verification

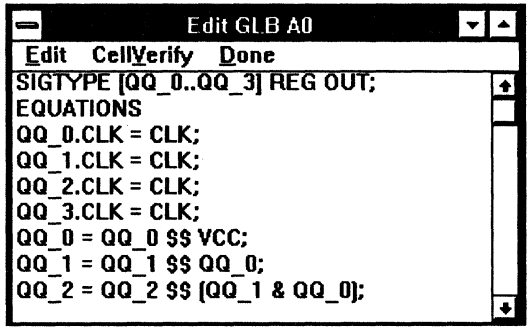
Cell Verification is performed as the design is entered in the I/O cells and GLBs. Design Verification is performed after all the I/O cells and GLBs have been entered and verified.

Cell Verification

To verify an individual cell:

1. Select *Cell* from the Main Menu.
2. From the *Cell* menu, select the *Edit* menu item. The Cell Edit Window appears as shown in figure 6-9.

Figure 6-9 Cell Edit Window



3. As the logic is entered in each I/O cell and each GLB, select *CellVerify* to verify the I/O cell or GLB.
4. When the I/O cell or GLB is verified, press *Done*, and go on to the next cell.
5. Repeat this process until all I/O cells or GLBs are entered and verified.

Cell verification performs the following five operations:

- Syntax Check
- Cell Design Rule Check
- Logic Minimization
- Logic Mapping
- Netlist Update (in memory)

The Syntax Checker reviews the logic for valid syntax, arguments, and appropriate punctuation. It then converts the ASCII format into the Logic Design Entry (LDE) internal binary format. This binary format is used by the rest of the programs.

The Cell Design Rule Checker checks the entered design against the physical limitations of the cell (GLB or IOC) and issues error messages, if any of the limits are exceeded. Table 6-3 shows a summary of the limits which are checked.

After the logic entry has been verified by the syntax checker, it is mapped into the device with the fitter. The fitter first reduces the raw equations to minimize the number of product terms. The fitter then fits (or maps) the resulting Sum of Products terms into the hardware. If the design does not fit in the cell, the software issues error messages indicating the problem. The software offers an override command NOMIN, which forces the fitter to bypass the equation minimization for special applications and for combinatorial logic hazard protection.

Table 6-3. Cell Design Limits

Inputs to GLB	18
Clock inputs to GLB	1
Product term clocks in GLB	1
Product term resets in GLB	1
Registers used in a GLB	4
Product term output enables generated in GLB	1

After the cell has been successfully verified, the program updates the global netlist that is in memory. When all cells have been verified, a netlist is generated containing all the nets within the design.

Design Verification

Design verification is a process of globally verifying all of the GLBs and I/O cells in the device. It also checks for any disconnected signals.

To verify a design:

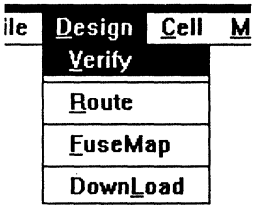
1. Select *Design* from Main Menu
2. Select *Verify* from *Design* Menu

Design Verify performs the following operations:

- ❑ Macro Verification on Unverified Macros
- ❑ Cell Verification on All Unverified Cells
- ❑ Global Design Rules and Connectivity Check
- ❑ Netlist Update (in Memory)

The Design Menu is shown in figure 6-10.

Figure 6-10. Design Menu



The process of verification of unverified Macros and cells updates the .LIF file in memory which is required for completion of the Global Design Rule check.

The Global Design Rule check searches for dangling nets and duplicate pin names, then compares the maximum number of resources needed versus the resource availability in the target device.

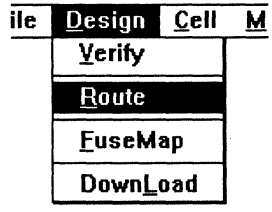
The .LIF file is created at the end of this process for use by the Router.

Lattice Place and Route

Lattice Place and Route (LPR) options are discussed in the following text. For a complete explanation of all the options available within LPR, refer to the pLSI/ispLSI Development System User Manual.

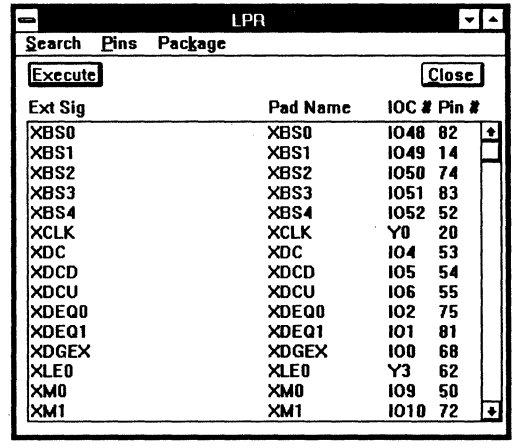
After the design has been successfully verified, the next step is to map the logic to the device. This task is accomplished from the LPR window which is activated by selecting the *Route* menu item under the *Design* menu (see figure 6-11 and figure 6-12).

Figure 6-11. Route Option



The LPR reads the design netlist that is created by the *Design Verify* process and a device file. The available resources for the chosen device are read from the device file by the LPR.

Figure 6-12. LPR Window



6

The LPR window has three menus available:

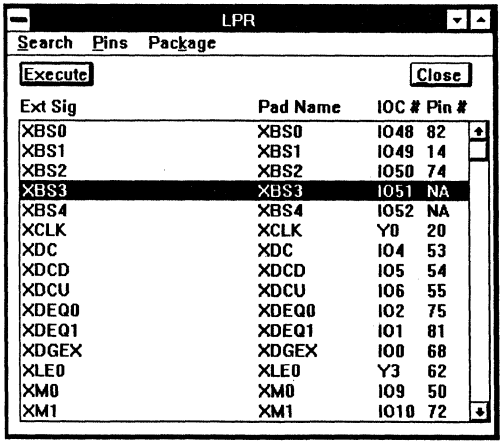
- *Search*
- *Pins*
- *Package*

There is also an *Execute* button which invokes the LPR software with the pin settings that are shown in the list box below the Execute and Close buttons. The *Close* button closes the LPR window and then displays the main window as it appeared before the LPR window was opened.

When a design is first routed the pins should not be fixed because this gives the LPR a greater chance to route successfully. If logic changes are needed, and the pins are not fixed, the router can reassign the pins. If a pin has not been assigned, the Pin # is called NA for Not Assigned (see figure 6-13).

After the design has been routed, the designer may want to make changes to the logic. If the pin assignment is not fixed, the LPR may reassign the pins.

Figure 6-13. LPR with an NA Pin



Fixed Pin Assignments

When fixed pins are required to avoid pinout changes, two methods are available. The first method is to define the pin number. This is done as the logic is entered in an I/O Cell, Clock Input, or Dedicated Input. This information is read and passed to the LPR. Example 6-5 shows the syntax for fixing a pin in an I/O Cell. The number after the keyword LOCK tells the LPR the pin number on which to fix the signal.

Example 6-5. Lock Usage

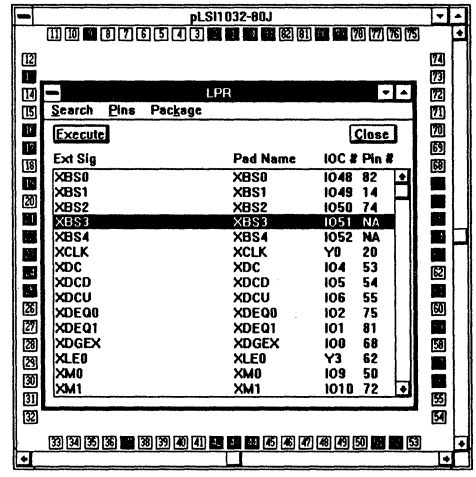
```
XPIN IO XQ7 LOCK 41;
OB11  (XQ7,Q7);
```

The second method is to select the Ext sig, Pad Name, or I/O number from the list by using the following steps:

1. Single click the left mouse button on the *Package* menu item. A diagram of the device pins appears. The pins that are grayed have not been selected and are available as shown in figure 6-14.
2. Select a Signal or an I/O Cell.
3. Single click on the pin number selected to lock the signal or I/O Cell to. The pin will no longer be gray and the pin number will appear in the Pin # column.

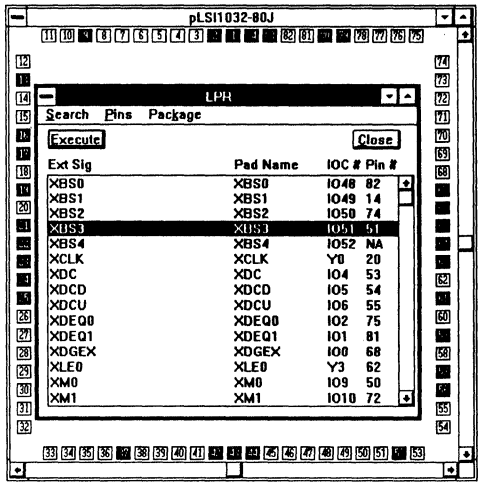
The router can now be executed at any time. Unfixed pins are automatically placed by the LPR. By selecting a signal in the signal list of fixed pins, the designer can unlock a pin and fix the signal to another pin.

Figure 6-14. LPR Window with Selected Pin and Package Menu



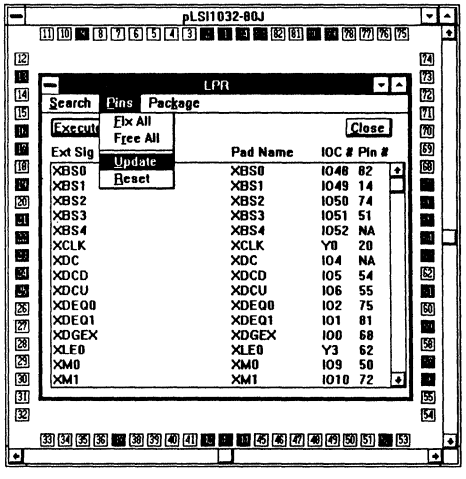
Under the *Pins* menu, two of the options are: Reset and Accept. Reset sets the pin assignment to whatever it was at the last Save or Accept. Accept uses the current pin settings as the default. This feature is valuable when attempting multiple routes and different pin assignments.

Figure 6-15. LPR Window with Pin Assigned



If a route is attempted and it is unsuccessful, the pins can be reset to the last setting that was saved. Another solution for an unsuccessful route is to unlock pins one by one and try to route the design again, until the route is successful. After the route is successful, an Accept is performed, and incremental routes can be completed without changing the pinout.

Figure 6-16. LPR Window with Pins Menu



Generating the Programming File

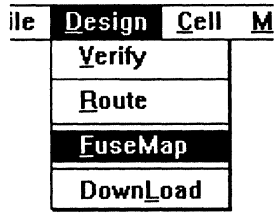
After successfully routing the logic design, it is ready for programming file generation. The programming file is a JEDEC file required for programming the pLSI and ispLSI devices.

The programming file generator reads the LPR database and uses this information to generate a programming file. The programming file is in ASCII format and can be read by any text editor. The programming file name is the design_name.JED.

There is only one option within the programming file generator, the security cell. The security cell has two options: on or off. With the security cell turned on, it is impossible to read the programming file information out of a programmed device. This is an important feature for systems with proprietary logic.

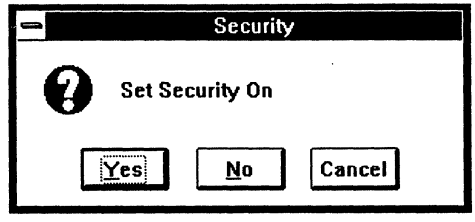
To generate a programming file, select the *FuseMap* menu item from the *Design* menu (see figure 6-17).

Figure 6-17 FuseMap



The programming file generator prompts for a selection of the security mode (see figure 6-18).

Figure 6-18 Security Cell



The programming file is automatically created. When it is finished, the message window prompts the designer to invok the Down Load option.

6

Down Load to a Device

There are two methods to program a device. One method is to transfer the programming file to a device programmer. The second method is to use the ispLSI version of the device and program the part directly, on the application board.

This section contains information about:

- Options for Programming a Device
- Using the Device in-system programming

Programming Using a Device Programmer

The ispLSI and pLSI devices can be programmed in a device programmer. The programming file, generated from the Fusemap generator, is transferred to the programmer in a method which depends on the programmer used, following the manufacturer's recommendations.

in-system programming

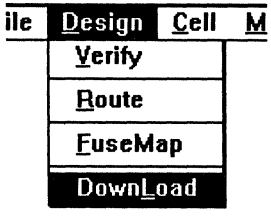
Lattice ispLSI devices have a very advanced and unique feature that allows programming and reprogramming the device in the system without removing the device from the board. This feature can eliminate board rework, redesign and layout of PCB and significantly reduce time to market. Enhancements and changes to logic can be accomplished without removing the ispLSI device from the PCB.

The in-system programming is accomplished using the Lattice isp 5-wire interface.

The Download Function

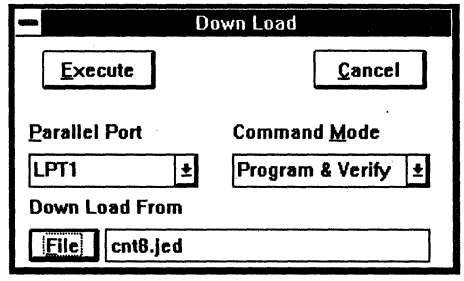
Select *DownLoad* in the Design menu as shown in figure 6-19 and the programming window will appear.

Figure 6-19. Design Menu, Down Load



This window allows selection of the LPT port that the download cable is connected to, the JEDEC file to be programmed and programming modes such as Program,Verify or Program & Verify. The default mode is Program & Verify and the default file is the current design programming file. The pDS software will automatically determine if the Download cable and device are connected correctly and which LPT port they are connected to. In addition, the download operation provides a warningin the isp mode if the device does not have power enabled. Figure 6-20 shows the in-system programming window.

Figure 6-20. in-system Download Window



Report Files

The pDS software has two methods of supplying the user with information. The first method is the message window (see figure 6-4 in this Section) which prompts the user with information, errors and the next step in the design process. The second method of supplying the user with information is the report files. This section will describe each report file and the information contained in each report. The report files that will be described are:

1. <design_name>.rp1 Design Verification Report
2. <design_name>.rp2 GLB Report
3. <design_name>.rp3 Resources Report
4. <design_name>.rp4 Netlist Report
5. <design_name>.rp5 External Pins Report
6. <design_name>.rp6 Routing Report

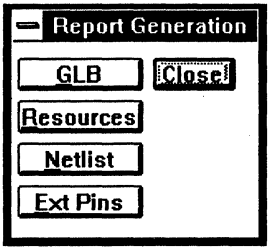
Each report has a header at the top of the file with the time the report file was generated, the version of the software that generated the file and the name of the report file. Example 6-6 shows the header section.

Example 6-6. Report Header

```
Mon Oct 21 09:48:04 1991
Counter.rpl generated using Lattice pDS V1.00.21
```

Reports 2 - 5 also have a second header section that lists the information from the Setup menu as well as the "Verify Status", the "Route Status" and the "Data Used" to generate the report, Pre-route or Post-route (see example 6-7). These 4 report files are optional and can be generated by selecting the report menu option (in the *File* menu) and clicking on the button for the desired report (see figure 6-21).

Figure 6-21. Report File Menu



Example 6-7. Second Report Header

```
Part: pLSI1032-80J
Design Name:  COUNTER
Design Revision:  1.00
Author:
Project Name:
Description:  This design is a vehicle for showing the report features
Verify Status:  Complete
Route Status:  Complete
Data Used:  Pre-Route
```

6

Design Verification Report

The Design Verification report, design_name.rp1 is generated automatically after a design verify as shown in example 6-8. This report lists any GLB or IOC that had not been previously verified and if the GLB or IOC was successfully verified. The report will also specify if the design verification was successful and will include any warning messages.

Example 6-8. Design Verification Report

```
Mon Oct 21 09:48:04 1991
COUNTER.rpl generated using Lattice pLSI V1.00.21
*** Begin Design Verify Message Log ***
Verify Design - COUNTER.LIF
LDE: Begin verification
Verifying cell B1.
Verifying cell A1.
Verifying cell C5.
Verifying cell B5.
Verifying cell IO55.
LDE: Successful verification
Successful Verification.... Ready For Routing!
 *** End Design Verify Message Log ***
```

GLB Report

The GLB report design_name.rp2 is always generated with the pre-route information. A full report is listed in example 6-11. This report is on a GLB basis, it lists the inputs, outputs and clock to each GLB with the signal names. The report also shows the "AND" usage and the output slot usage.

The "AND" usage portion of the report is presented in example 6-9 which shows the product term configuration with a 1 for each PT position that is used. The PTs shown are in the default configuration of 4, 4, 5, 7. The PTs are shown from PT0 on the left to PT19 on the right. The vertical lines indicate the partition of the product terms.

Example 6-9. And usage

```
AND Usage for GLB:
   A5                :1000|0000|00000|0000000
```

The Output Slot Usage portion of the report shows the path through the GLB by indicating which resources were used. Example 6-10 indicates that a single product term is used and the rest of the resources are not used. The single PTs are listed as PT0, PT4, PT8 and PT13 which are from the AND array. The rest of the notations refer to Outputs O3 to O0. The "OR" indicates the Product Term Sharing Array (PTSA), the XOR is the internal Hardware Exclusive-OR gates, the "4PTBYPASS" is the four product term bypass which bypasses the PTSA and XOR, and the REG is the outputs of the GLB flip-flops.

Example 6-10. Output Slot Usage

```
OUTPUT Slot Usage for GLB:
   A5                : SINGLEPT1000 OR0000 XOR0000 4PTBYPASS0000 REG0000
```

Example 6-11. GLB Report

```
   Mon Oct 21 10:31:06 1991
   COUNTER.rp2 generated using Lattice pDS V1.00.21
GLB Report
   Part:    pLSI1032-80J
   Design Name:   COUNTER
   Design Revision:     1.00
   Author:
   Project Name:
   Description:   This design is a vehicle for showing the report features
   Verify Status: Complete
   Route Status:  Complete
   Data Used:     Pre-Route
STATISTICS FOR GLB B1
GLB Input List:
   I0    : A4
   I1    : A3
   I2    : A2
   I3    : SEL1
   I4    : SEL0
GLB Output List:
   O3     : A5
AND Usage for GLB:
   A5          : 1000|0000|00000|0000000
OUTPUT Slot Usage for GLB:
   A5          : SINGLEPT1000 OR0000 XOR0000 4PTBYPASS0000 REG0000
STATISTICS FOR GLB A1
GLB Input List:
   I0    : A3
   I1    : A2
   I2    : SEL1
   I3    : SEL0
```

```
GLB Output List:
   O3    : A4
AND Usage for GLB:
    A4          : 1000|0000|00000|0000000
OUTPUT Slot Usage for GLB:
    A4          : SINGLEPT1000 OR0000 XOR0000 4PTBYPASS0000 REG0000
STATISTICS FOR GLB C5
GLB Input List:
   I0    : A2
   I1    : SEL1
   I2    : SEL0
GLB Output List:
   O3    : A3
AND Usage for GLB:
    A3          : 1000|0000|00000|0000000
OUTPUT Slot Usage for GLB:
    A3          : SINGLEPT1000 OR0000 XOR0000 4PTBYPASS0000 REG0000
STATISTICS FOR GLB D3
GLB Input List:
   I0    : SEL1
   I1    : SEL0
GLB Output List:
   O3    : A2
AND Usage for GLB:
    A2    : 1000|0000|00000|0000000
OUTPUT Slot Usage for GLB:
   A2         :SINGLEPT1000 OR0000 XOR0000 4PTBYPASS0000 REG0000
STATISTICS FOR GLB B5
GLB Input List:
   RESET : RESET
   I0    : A5
   CLK0  : INTCLK
GLB Output List:
   O3    : FF1
AND Usage for GLB:
    FF1         : 1000|0000|00000|0000000
OUTPUT Slot Usage for GLB:
    FF1         : SINGLEPT1000 OR0000 XOR0000 4PTBYPASS0000 REG1000
```

Resources Report

The Resources report, *design_name*.rp3 is a listing of the number of GLBs, IO cells, dedicated inputs, external nets, GLB clocks and I/O Cell clocks used in the design as shown in example 6-12. This report can be generated before the logic has been routed as well as after the routing is complete. This will enable the user to identify how many extra GLBs the LPR required to accomplish the route and how many GLBs are available for additional logic.

Example 6-12. Resources Report

```
COUNTER.rp3 generated using Lattice pDS V1.00.21
Global Resource Utilization Summary
Part: pLSI1032-80LJ
```

Example 6-12. Resources Report (continued)

```
   Design Name:      COUNTER
   Design Revision:      1.00
   Author:
   Project Name:
Description:    This    design   is   a vehicle for showing the report features
   Verify Status: Complete
   Route Status:  Not Done
   Data Used:     Pre-Route
Number of GLB cells used:          5
Number of IO cells used:           3
Number of dedicated inputs used:   0
Number of external nets used:      4
Number of GLB clocks used:         1
Number of IO clocks used:          0
```

Netlist Report

The Netlist Report*design_name.rp4* lists in a tabular format, the net name, the GLB where the net is coming from and where it was originally, the destination GLB(s) and their original GLB locations and the fanout of the net. This information may be very useful for repartitioning a design to lower fanouts and possibly increase utilization. This report only has pin to pin information and does not include any nets internal to the GLB or IOC (refer to example 6-13).

Example 6-13. Netlist Report

```
COUNTER.rp4 generated using Lattice pDS V1.00.21
Netlist Report
   Part:    pLSI1032-80J
   Design Name:          COUNTER
   Design Revision:      1.00
   Author
   Project Name:
Description:    This   design   is   a vehicle for showing the report features
   Verify Status: Complete
   Route Status:  Complete
   Data Used:     Post-Route

Net Name      Source Pin       Destination Pin(s)      Fanout
              cell-pin         cell-pin
              (org cell)       (org cell)

!XRESET       RST                                        1
!RESET        RST              C0-RESET (B5)             1

Net Name      Source Pin       Destination Pin(s)      Fanout
              cell-pin         cell-pin
              (org cell)       (org cell)

A5            B4-O0 (B1)       C0-I15 (B5)               1
A3            A4-O1 (C5)       B0-I1 (A1)                2
                               B4-I1 (B1
```

Example 6-13. Netlist Report (Continued)

```
A4            B0-O2 (A1)      B4-I2 (B1)                1
FF1           C0-O0 (B5)      IO32-IR (IO26)            1
INTCLK        Y0              C0-CLK0 (B5)              1
SEL0          IO4-O (IO55)    A4-I4 (C5)                4
                              B0-I4 (A1)
                              B4-I4 (B1)
                              A0-I4 (D3)
SEL1   IO3-O (IO60)           A4-I3 (C5)                4
                              B0-I3 (A1)
                              B4-I3 (B1)
                              A0-I3 (D3)
A2       A0-O0 (D3)           A4-I0 (C5)                3
                              B0-I0 (A1)
                              B4-I0 (B1)
```

External Pin Report

The External Pin report, design_name.rp5 may be generated after a successful routing session (see example 6-14). This report will list the Pin Number, Pad Name, Fixed (yes or No) and the Pin Type (input, output or bidirectional).

Example 6-14. External Pin Report

```
   COUNTER.rp5 generated using Lattice pDS V1.00.21
External I/O Pin Report
   Part:    pLSI1032-80LJ
   Design Name:          COUNTER
   Design Revision:      1.00
   Author:
   Project Name:
Description:   This   design is a vehicle for showing the report features
   Verify Status:        Complete
   Route Status:         Complete
   Data Used:            Post-Route
    Pin            Pad Name      Fixed       Pin Type
    Number
    29             XSEL1         No          Input
    30             XSEL0         No          Input
    68             XFF1          No          Output
    20             EXTCLK        No          Input
```

Routing Report

The Routing report, *design_name.rp6* is an automatically generated report. This report file is only generated if the routing was unsuccessful. The report has two sections. The first section lists each GLB output and each IOC used and indicates if the output routed or not. The second section is the Routing Failure Report, which lists each GLB output signal, the fanout, the destination(s) of the signal and which signals are conflicting.

Clock Distribution Network

In previous sections, the architecture of the Clock Distribution Network was explained. As mentioned, there are four external clock pins (except in pLSI 1016 and ispLSI 1016, which have three clock pins). Internally there are three global GLB clocks and two I/O clocks for use with input registers and input latches. The following topics are described:

- Locking Clock Pins
- Global Clocks
- Product Term Clock
- Global I/O Clocks with Input Registers
- Clock Polarity

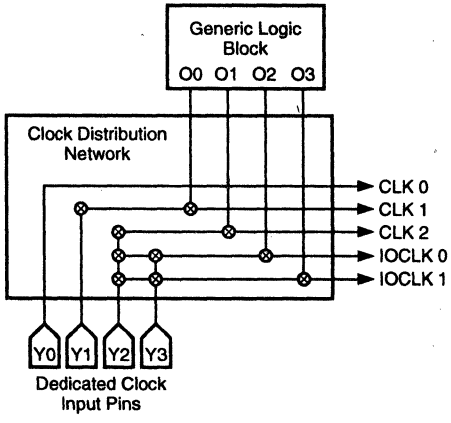
Locking Clock Pins

Three internal clock lines provide global synchronous clock distribution to the GLBs and two clock lines provide distribution to the I/O cells. The global clock lines are available to all the GLBs in a device and offer minimum skew. Like the GLB global clock lines, the global I/O clock lines are available to all I/O cells.

In designs with more than one external clock, assign and lock the most critical clock to the Y0 pin, since CLK 0 is hardwired and CLK 1 and CLK 2 are not hardwired (see figure 6-22).

Another reason for locking clock pins is to assure the order of the clock signals. The source of a clock signal is associated to it's destination by signal name. The Design Verify reads the GLBs first, and then the I/O cells.

Figure 6-22. Clock Distribution Network



As the design verify procedure finds clock signals, it assigns the external clock pins and the associated internal clock lines. The order it assigns clocks is shown in table 6-4.

Table 6-4. Clock Pins and Associated Clocklines

Order	Devices	Clocks
Y0 to Y2	1024 – 1048	GLB Clocks
Y2 to Y3	1024 – 1048	I/O Clocks

Clock pins should be locked to assure that the same clock pins are assigned when doing a redesign or an enhancement. For example, in figure 6-22, Y2 can be either a GLB or an I/O clock pin on the pLSI 1024, 1032 and 1048 and ispLSI 1024, 1032 and 1048. Locking clock pins does not effect the routability of the design.

Example 6-16 illustrates the syntax required to specify clocks pins and to lock them to a particular pin.

Example 6-16.

```
XPIN CLK X_1MHZ LOCK 20
IB11 ( _1MHZ, X_1MHZ ) ;
```

where:

CLK — indicates that the input signal is from an external clock pin, as opposed to an I/O or dedicated input.

20 — is the pin number and it overrides the pin placement entered using the Graphical User Interface (GUI). The pin number can also be locked during Place and Route using menu driven commands.

X_1MHZ — is the pin name.

_1MHZ — is the net name.

Global Clock

As shown in the architecture section and in figure 6-23, there are four registers in every GLB. The four registers have a common clock. The clock selection is done on an individual GLB basis. The clock signal is only declared once in each GLB. Every GLB selects from four different clock sources. Three of these clocks are global GLB clocks. The fourth is a product term clock which is described later.

Example 6-17 shows a clock declaration using equations:

Example 6-17.

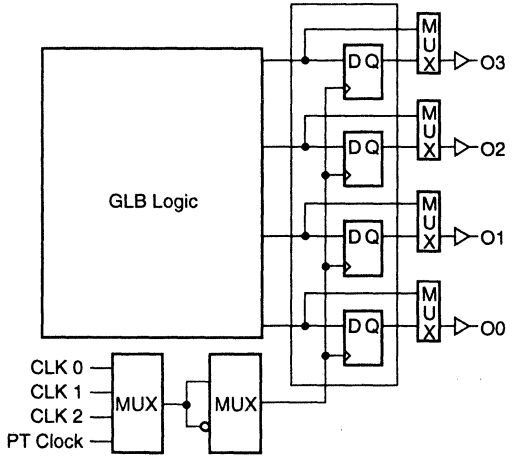
```
SIGTYPE QXX REG OUT;
EQUATIONS
    QXX.CLK=_1HZ;
    QXX=((ADDR1 & ADDR2) # (EN4 & CS));
END
```

where:

.CLK Declares that the clock comes from one of the three global GLB clocks, CLK0-CLK2.

_1HZ Is the clock signal name

Figure 6-23. GLB, Registers and Clocks



Example 6-18 shows clock usage in a Macro.

Example 6-18.

```
SIGTYPE _10HZ REG OUT;
SIGTYPE Bell REG OUT;
    //pinorder (Q, J, K, CLK)
    FJK11 (_10HZ, SIG2, SIG5, _1MHZ);
    FJK11 (BELL, timer, alarm, _1MHZ);
```

Product Term Clock

The pLSI and ispLSI families also generate and use asynchronous clocks. These clocks are known as product term clocks (PT Clock). A PT Clock is a clock that is produced within a GLB and is a function of the signals from the Global Routing Pool (GRP). The PT Clock is connected to PT12 and does not leave the GLB (see figure 6-23).

The syntax for using an asynchronous or PT Clock is slightly different from the syntax for using a global clock. Examples 6-19 and 6-20 illustrate logic using a product term clock.

Example 6-19.

```
SIGTYPE HR0 REG OUT;
SIGTYPE HR1 REG OUT;
EQUATIONS
    HR.PTCLK=_1PPHR;
    _1PPHR=MIN50 & MIN9;
    HR0= (HR0 & !LD) $$ (D0 & LD);
    HR1= (HR1 & !LD) $$ ((D1 & LD)
       # (HR0 & !LD) # GND));
END
```

where:

.PTCLK Declares that the clock is a PT Clock.

Example 6-20.

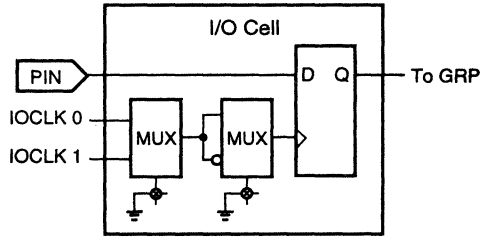
```
SIGTYPE _10HZ REG OUT;
    //pinorder FJK11 (Q, J, K, CLK)
     FJK11(_10HZ, SIG2, SIG5,_1MHZ.PTCLK);
EQUATIONS
    _1MHZ=SEC0 & !SEC1 & !SEC2 & !SEC3;
END
```

Global I/O Clocks with Input Registers

Input functions in I/O cells are registered, latched or combinatorial functions. As shown in figure 6-24, there are two global I/O clocks for each registered input. The designer can individually select which of the two global I/O clocks controls the register or latch. Input, output and bidirectional functions are only available in Macro form. Macro selection is described in the Macro Usage section.

Only the correct signal name needs to be defined, i.e. I/O CLOCK, when describing the I/O cell functionality. In figure 6-24, the I/O cell is configured as a registered input using an input register Macro named ID11 with a global I/O clock named IOCLOCK.

Figure 6-24. Global I/O Clocks



Clock Polarity

A multiplexer is included in both GLBs and I/O cells. The multiplexer selects between the rising edge or the falling edge clock. The clock can be inverted or have its polarity changed by adding an "!" before the clock signal name in equations and Macros. This is done for specific I/O cell clocks, GLB clocks, or for global clocks before they reach their destinations.

Macro Usage

A Macro is a commonly used function such as an AND Gate, or an operation such as an adder. Macros can be simple or can be combined with other Macros to build complex Macros. A library is a file that contains a group of Macros.

Libraries and Macros are described in the following topics:

- System Macro Library
- Macro Types
- Library Types
- Selecting and Adding a Library
- Selecting and Editing a Macro
- Macro Syntax

System Macro Library

The system Macro library (latsysm.lib) contains more than 225 Macros. These Macros range from inverters and buffers to arithmetic functions, including multiplexers and demultiplexers. For a detailed list of the available Macros, refer to the Lattice Macro Library Manual.

Macro Types

There are two types of Macros

- Soft
- Hard

Soft Macros are represented by logic equations. These equations are in ASCII format so the designer can read them. A compiled version of the equations is read by the pDS software. The compiled version is generated when a Macro is verified and saved.

Hard Macros are only available in the system library and are similar to soft Macros except they have no read and write permission. They cannot be edited or read. Hard Macros are compiled with external software. They are optimized to take advantage of enhanced features within the architecture and for performance. Hard Macros are pre-fitted which allows faster placement and routing of the logic. All interface logic, inputs, outputs and bidirectional functions, are hard Macros.

Library Types

There are also three types of libraries:

- System
- User
- Design

The system library is named latsysm.lib and is provided with the entry level software. This file is located in the following path:

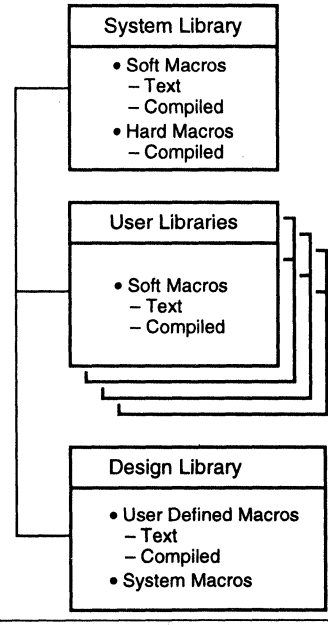
lattice\macrolib\latsysm.lib

The system library contains hard and soft Macros.

User-libraries are libraries that users create. These libraries, contain soft Macros. The pDS software supports multiple user-libraries, which can be placed in any directory.

The design library is part of the design. It includes the soft Macros created and saved in the user-library and a table that lists the Macros selected from the system library. The design library is in both text and compiled formats. The compiled format is read by the software and the text format is provided so users can read or edit existing Macros to create other Macros.

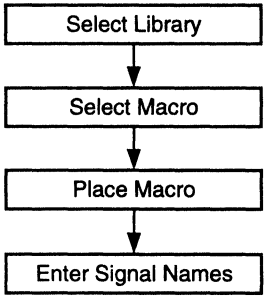
Fig. 6-25. Library Structure Diagram



Selecting and Adding a Library

Use the Library Menu to select a library. The default library is the system library (Latsysm.lib)if the design is empty. To select a Macro from the library other than the system library, it is necessary to first selectthe library containing the Macro. Only one library can be active at a time.

Figure 6-26. Flow Chart for Using Macros

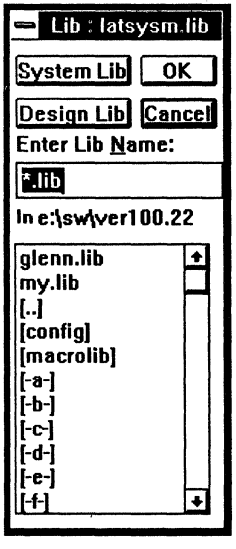


The library menu has two quick selection menu items:

- *Design Lib*
- *System Lib*

There is no user-library selection button, because the drive, path, and file must be specified to access a user-library (see figure 6-27).

Figure 6-27. Library Menu

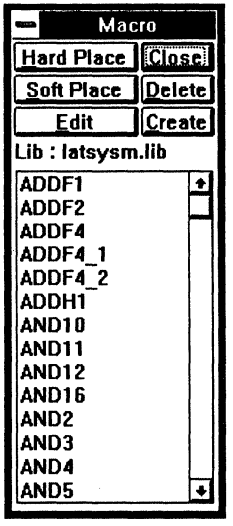


Use the Setup window to add or delete a library in the search list. Specify the device type, the project name, revision number and search order. When a library is deleted from the search order, only the path is deleted, not the library (see figure 6-27).

Selecting and Editing a Macro

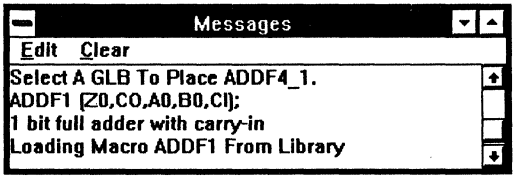
To select a Macro, click on the *Macro* menu item to display the Macro menu (see figure 6-28).

Figure 6-28. Macro Menu



Use the Macro menu to *Edit*, Place, *Create* or *Delete* Macros from the selected library. Click once on a Macro and the functional description appears in the message window (see figure 6-29).

Figure 6-29. Macro Message Window



To place a selected Macro into a GLB or an I/O cell, click on the *Place* button. If a GLB or an I/O cell is not selected the message window prompts the designer to select one. If the Macro is a multiGLB Macro, the message window

6

prompts the user to select another GLB or I/O cell until all the parts have been placed. MultiGLB Macros are placed in reverse order, i.e. _2 is placed, then _1. Once the Macro is placed in the design, the Macro becomes part of the design library until the design is deleted. The placed Macro retains the Macro name, required signal order, and the syntax (see figure 6-30).

Replace the input and output Macro names with signal names. A multiGLB Macro can also be placed by, clicking on the *Hard Place* button, the top level Macro will then be placed into the selected cell. When a *CellVerify* is selected, the pDS software will automatically reserve the necessary number of cells for the Macroparts. The reserved cells cannot be edited. The advantage of the *Hard Place* is, that it is not necessary to worry about the signal names in the Macroparts, as all the Macroparts will be placed for the user. Depending on the Macro being used, it may be more efficient to place the Macroparts individually so that the unused portion of the GLB can be used.

The Lattice program checks the Macro when a cell is verified. The program searches for the Macro in the design library first and if it is not found, the search order specified in the setup is used. The default for the search order is the system library.

Figure 6-30. Placed Macro

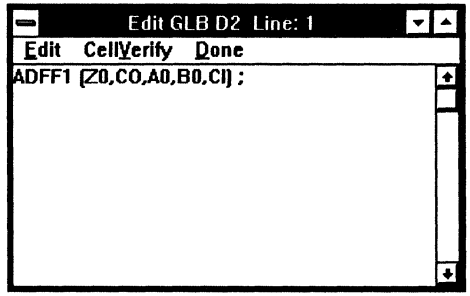
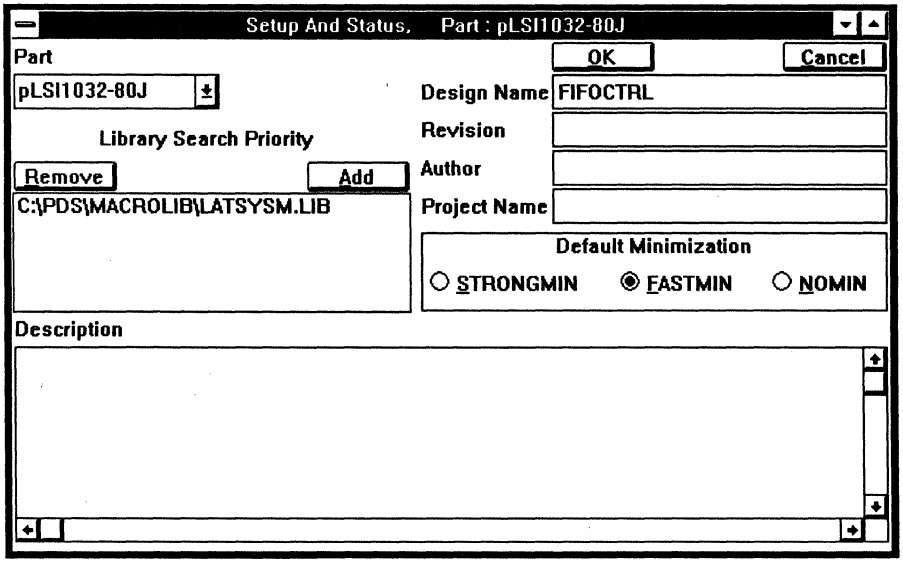


Figure 6-31. Setup and Status Window



Macro Syntax

To create a Macro, use the following syntax:

```
[{ MACRO <Macro_name> (<param_name> [{,<param_name>} * ]);
  [MACROTYPE [x|rx|rwx];]
  [MACROGROUP <Macro_group_name> ;]
  [MACROCOMMENT <comment> ;]
    [{ CONSTANT <constant_name> <constant_value>;} * ]
    [{ SET <set_name> [<param_name> [{,<param_name>} * ]] ;}*]
    [{ SIGTYPE <param_name> {<signal_attribute>}*;}*]
    [{<nested_macro_name> ([!]<param_name> [{,[!] <param_name>}*]);}*]
    EQUATIONS[;]
      {[!]<param_name>.[<dot_extension>] = <expression>;} *
    END [;]]
END [;] }*]
```

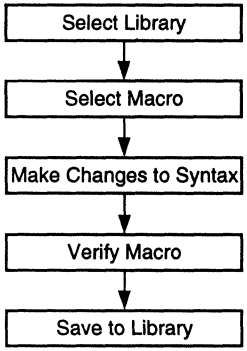
Where:

<X>	X is required
[X]	X is optional
{X}*	X is repeated

Macro	Indicates the beginning of a Macro definition, the usage, and the signal order.
Macrotype	Declares the type of Macro. The defaults are RWX.These are highlighted above, in the full Macro syntax.
Macrogroup	Declares which functional group the Macro is in, i.e. bincntr, the group name for binary counters.
Macrocomment	Is a comment line that is displayed when a Macro is selected from the library. The comment is typically a functional description, but can be anything you want to display.
Macropart	Declares that the Macro being used is a multiGLB Macro. Each Macropart is a separate GLB. Multicell Macros can only be GLB type Macros.

Constant, Set, Sigtype and Equations have the same functions as described in Entering the Design section. Remember, user-defined Macros are optional.

Figure 6-32. Flow Chart for Editing Macros

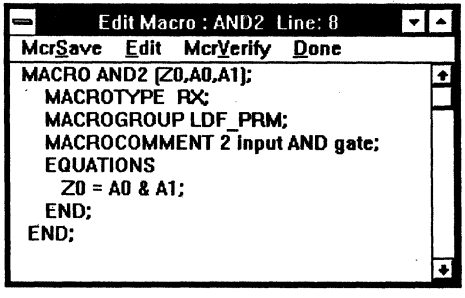


The simplest way to create a new Macro is to edit an existing Macro and save it in the design or user-library. For example, use the following steps to edit a 2 input AND gate (AND2), save it as a 4 input AND gate, and rename it AD4.

1. Click on the *Edit* button in the Macro window and make the necessary changes to the selected Macro.
2. Make the following changes to the syntax:
   - Change Macro_name from `AND2` to `AD4`
   - Change Macrotype from `RX` to `RWX`
   - Change Macrogroup from `LDF_PRM` to `Gates`
   - Change Macrocomment from `2 Input AND gate` to `4 Input AND gate`
   - Change Equations from `Z0 = A0 & A1` to `Z0 = A0 & A1 & A2 & A3 & A4`

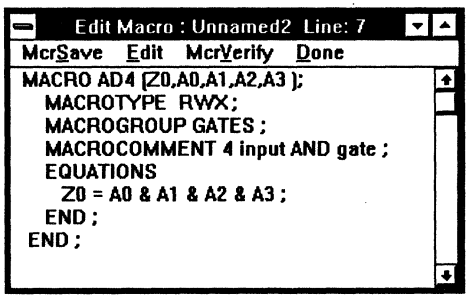
As stated previously, the default Macrotype for user defined Macros is RWX permission. If a Macro is defined without the RX or X permission option, then a *McrSave* is done the pDS software will provide a warning that the Macro is a write once Macro. This means the Macro cannot be edited or deleted. The only way to delete a write protected (write once) Macro is to delete the library!

Figure 6-33. Editing a Macro



The changes are shown in figure 6-34.

Figure 6-34. Unnamed Macro

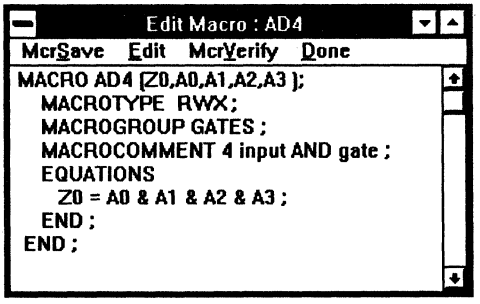


When all the changes have been made, the Macro is verified. *McrVerify* performs the same function as *CellVerify* does in a GLB or IOC. It checks for syntax and whether all the logic described will fit in the GLB. A temporary name is assigned automatically when any change is made to a Macro. The temporary name is Unnamedx, where *x* is a number.

Macros must be successfully verified before they can be saved by their Macro name. If a Macro verification is unsuccessful, the Macro keeps its temporary name, which is assigned by the program, until it is successfully verified.

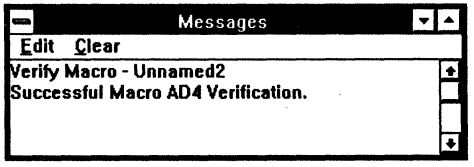
Figure 6-35 shows the Macro verified and unnamed1 changed to *AD4*.

Figure 6-35. Window Showing Verified Macro



The message window shows that the verification was successful.

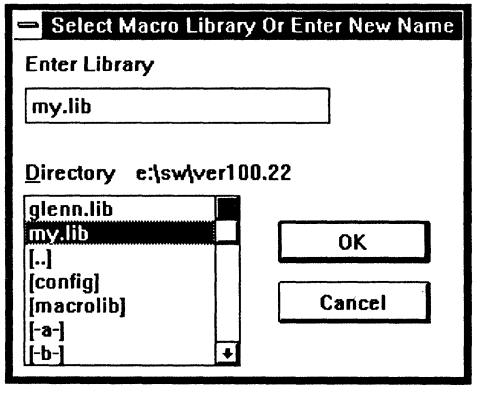
Figure 6-36. Message Window



After a Macro is successfully verified, the *McrSave* option is enabled.

McrSave saves user-defined Macros to user-libraries. User-defined Macros, are automatically saved to the design library. No two Macros should have the same name. If two Macros are functionally different and the names are the same, the system asks if this new version should be used. If the designer types yes, all the Macros with the same name are replaced with this Macro and the design may not function as intended (see figure 6-37).

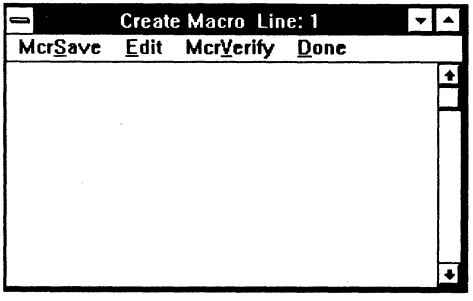
Figure 6-37. Saving a Macro



To close the Macro edit window, click once on the *Done* option. The done option also allows the designer to close a Macro that has not been successfully verified.

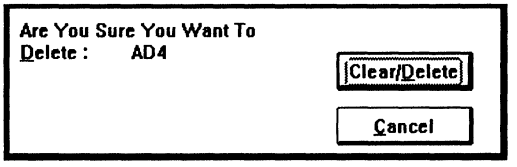
The Macro *Create* button does the same functions as Macro *Edit* except, the window is empty and the user must enter all the syntax manually (see figure 6-38).

Figure 6- 38. Macro Create Window



Macro *Delete* allows the designer to delete a Macro from a library. When the Macro *Delete* button is selected, a dialog box is displayed as shown in figure 6-39.

Figure 6-39. Macro Delete Window



A Macro is part of a file. If a Macro from the library is deleted, it cannot be recovered.

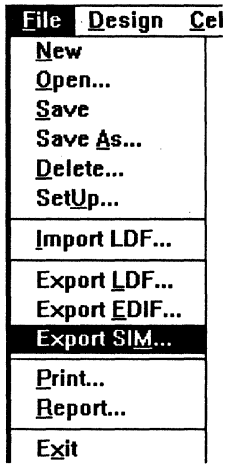
Logic Simulation of the Design

A design may be verified by programming and testing the device or testing the device in the system; or it can be simulated with the Viewsim logic simulator. This section will explain the process needed to do a simulation. For more details on Viewsim, please refer to the Viewsim documentation. This section assumes that the Viewlogic software has already been installed on the design platform. For installation of the Viewsim software, please refer to the sidebar of this section.

Netlist Formats

A netlist is a file that describes all the logic and connectivity of the design. A netlist can be generated for simulation from the pDS only after a successful route. The netlist can be in either Electronic Data Interface Format (.EDN) or a simulation (.SIM) file format. Either format can be generated for use in simulating a design. The *Export EDIF* and the *Export SIM* commands are under the *File* menu (refer to figure 6-40).

Figure 6-40. Export Commands



EDIF Format

When the menu item *Export EDIF* is selected, it prompts for the name of the EDIF file. The EDIF file can be used for interface to third party simulation programs.

Simulation Format

While using the pDS software, if the menu item *Export SIM* is selected, it prompts for the name of the SIM file. The file name will be the name of the design with a SIM extension. This file should be placed in project directory.

The .SIM file is an intermediate format. To convert this format to the viewsim, VSM format type the following command:

LSC2VL FILE_NAME

The .SIM extension is automatically looked for.

To start a simulation type the following:

VIEWSIM NETWORK_NAME

Simulation Command File

Commands may be entered interactively, or read from a command file. Command files are recommended since there are setup commands that must be executed every time Viewsim is started. All registers and latches must be reset at the beginning of a simulation and VCC must be defined as a logic 1. The command file description also instructs Viewsim on which outputs or internal nodes to watch, the stimulus for the inputs or internal nodes, the clock timing and the length of the simulation, to name a few of the options. For more details, refer to the Viewsim manual. The vertical bar, "|" in the first column signifies that line is commented out. The command file for the project should be placed in PROJECT_NAME\FILE_NAME.CMD, however it may be placed in any directory on the system. To execute a command file that is not in the project directory, the full path must be specified. The file can have any name however, if the file extension .CMD is used, the extension does not have to be entered when the command file is executed.

A log file is opened automatically when Viewsim is invoked and all simulation commands are written to this file. The line "log PROJECT_NAME.log" in the sample.cmd file can be modified to any name. If commands are entered interactively, this file may be edited to create a command file.

To execute a command file from within Viewsim, type the following:

EXECUTE FILE_NAME

Note: the output of the simulation will go to the display unless re-directed to a file. The sample command (.CMD) file should be edited. The third line should be changed to the name of the project. This will create a log file named PROJECT_NAME.LOG.

Installing Viewsim

This side bar describes the installation of the Lattice Macro Library with Viewlogic's Viewsim simulator. The library is furnished to provide design verification for the Lattice design editor.

Requirements for Operation

- IBM compatible 386 PC
- Viewlogic Viewsim version 4.1
- Viewlogic Viewsim wirelister for version 4.1
- Lattice Macro Library for Viewlogic
- 4 Megabytes of RAM (2 megabytes or more of extended memory. Viewsim requires 1M for every 5000 gates plus overhead).
- 2 Megabytes of Disk Space (Lattice and Viewlogic)
- Parallel Port for Security Block

Installation

The installation program will automatically check the system directory structure, modify the autoexec.bat file to include the environment variable and workview in the path, place all the files in the directories specified and modify configuration files necessary to reflect the library path.

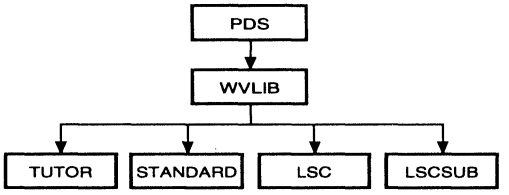
Set your current working directory to the root directory of the drive where the software will be installed. Place the installation disk in floppy drive A: and enter the following:

A:INSTALL

If a drive other than A: is used, substitute the correct drive name. The installation program will prompt the user for the directory in which to install the Viewlogic software. The installation program will also prompt the user for the directory in which to install the Lattice Macro Library for Viewsim (WVLIB). The default directory for the WVLIB is \PDS. The directory structure of the WVLIB is shown (see figure 6-41).

A copy of the Viewsim command file (.CMD), VIEWDRAW.INI file, and VIEWSIM.VAR file will be installed in the STANDARD directory. These three files in the STANDARD directory are for backup. Please refer to the appendices below for copies of the command file and the directory section from the VIEWDRAW.INI file. The VIEWDRAW.INI and viewsim.var files will also be installed in the WORKVIEW\STANDARD directory.

FIGURE 6-41. Lattice Macro Library directory structure for Viewsim



The following paragraphs are provided for information. only The install program performs these steps automatically.

Configuration

The sample VIEWDRAW.INI file installed with the WVLIB specifies the location of the Lattice library for the Viewlogic applications. A listing of the directory section is shown in appendix B. The Viewlogic applications also require an environment variable named WDIR to specify the WORKVIEW\STANDARD directory. This is usually done as shown below:

SET WDIR=C:\WORKVIEW\STANDARD

The sample files listed below assume the Viewlogic software was installed under C:\WORKVIEW. Use the actual pathname where the software was installed. In order to invoke the Viewlogic applications without specifying the complete pathname, add the c:\workview directory to the path command in the autoexec.bat file.

The Viewlogic applications use a DOS extender to access Extended memory on the PC. There are known problems when trying to run these programs in the Microsoft Windows environment. If the user runs the Viewlogic applications from within Microsoft Windows a message will be received complaining of insufficient memory. A simple but inelegant solution involves creating two sets of CONFIG.SYS and AUTOEXEC.BAT files. Use one for the Windows environment and the other for the Viewlogic programs. The PC will have to be reset whenever the environment is changed. Be sure to remove himem.sys from the Viewlogic config.sys file. A more sophisticated solution is to use a memory manager such as Quarterdeck's QEMM. By using QEMM it is possible to run the Viewlogic applications within the

Windows environment. However, Windows must run in the standard mode rather than enhanced mode.

Sample Viewsim command (.CMD) file

```
      log  PROJECT_NAME.log
| setup defaults:
| -cmdfile: do not print command file
| -watch:   do not display watch list, show simulation time
| time:     after every command that causes simulation time to
| forcex:   force built in registers to unknown state
      defaults -cmdfile -watch time forcex
|setup what errors are reported
| -spikes:  do not report spikes
| timing:   report timing errors such as set up and hold violations
|     report -spikes timing
| setup input and output vectors, define radix for the vectors, setup the
| watch list, setup the Viewwave file, set step size, define and start clock.
| vector commands allow shorthand labels for signals or sets of signals
| watch command determines which signals are displayed
| wave command links Viewsim to Viewwave enabling waveform display
      vector data d[3:0]
      vector control cd cs ps
      vector inout clk cd ps cs data
      watch clk cd ps cs data
      wave mydesign.wfm inout
| setup 20ns clock pulse
| step command sets the default simulation duration
| clock command specifies transitions for clock and other repeating signals
      step 10ns
      clock clk 0 1
| define VCC as 1 and setup the global reset signal.
| These are internal signals used in the Lattice Macros and must be
|  initialized
| each time Viewsim is started or restarted
| Units are specified in tenths of nanoseconds.
      h VCC
      wfm !XRESET 0=0 100=1
| define input stimulus
      after 40ns do (wfm control 0=0)
      after 40ns do (wfm data 0=0 (20ns=inc by 1)*16)
| print values to an output file for any change in state of signal in group
| inout
      break inout ? do (print >results.out)
| run the simulation for 18 cycles
| length of cycle is determined by clock command
```

Sample of Design Directories in Viewdraw.ini file

```
| Viewdraw initialization file for Version 4.1
| (c) Copyright 1985, 1991 Viewlogic Systems, Inc.
| Design Directories
| ----------
| Format: DIR [DirType(s)] DirPath (LibName)
| DirType: p or pw - primary / writable
| w - writable (read/write)
| r - read-only
| m or rm - read-only megafile
| DirPath: directory specification
| LibName: library name aka library alias or VHDL library name (optional)
| 32 characters or less. Must begin with a letter.
| examples:
|
| DIR [p] .
| DIR [r] /workview/wvlibs/74ls (vl74ls)
| DIR [r] /workview/wvlibs/builtin (builtin)
| DIR [w] under/development/alu (newalu)

DIR [p] .
DIR [rm] c:\LATTICE\WVLIB\LSC (lsc)
DIR [rm] c:\LATTICE\WVLIB\LSCsub (lscsub)
```

Section 1: Introduction to pLSI™ and ispLSI™

Section 2: pLSI and ispLSI Data Sheets

Section 3: GAL® Data Specifications

Section 4: pLSI and ispLSI Architecture

Section 5: pLSI and ispLSI Advantages

Section 6: pLSI and ispLSI Software Development Tools

Section 7: pLSI and ispLSI Design Optimization

GLB Partitioning 7-1

Placement and Routing 7-9

Device Utilization 7-17

Speed 7-21

Section 8: pLSI and ispLSI Application Notes

Section 9: pLSI and ispLSI Article Reprints

Section 10: pLSI and ispLSI Quality, Reliability & Programmer Support

Section 11: Sales Information

Design Optimization: GLB Partitioning

Design Optimization

In order to maximize performance, designers must understand how to optimize their designs for speed and utilization. This section will discuss how to partition the design into GLBs, how software keywords are used to enable different GLB features, and how the software works with the hardware features. Some design partitioning examples will be provided to help clarify the concepts discussed.

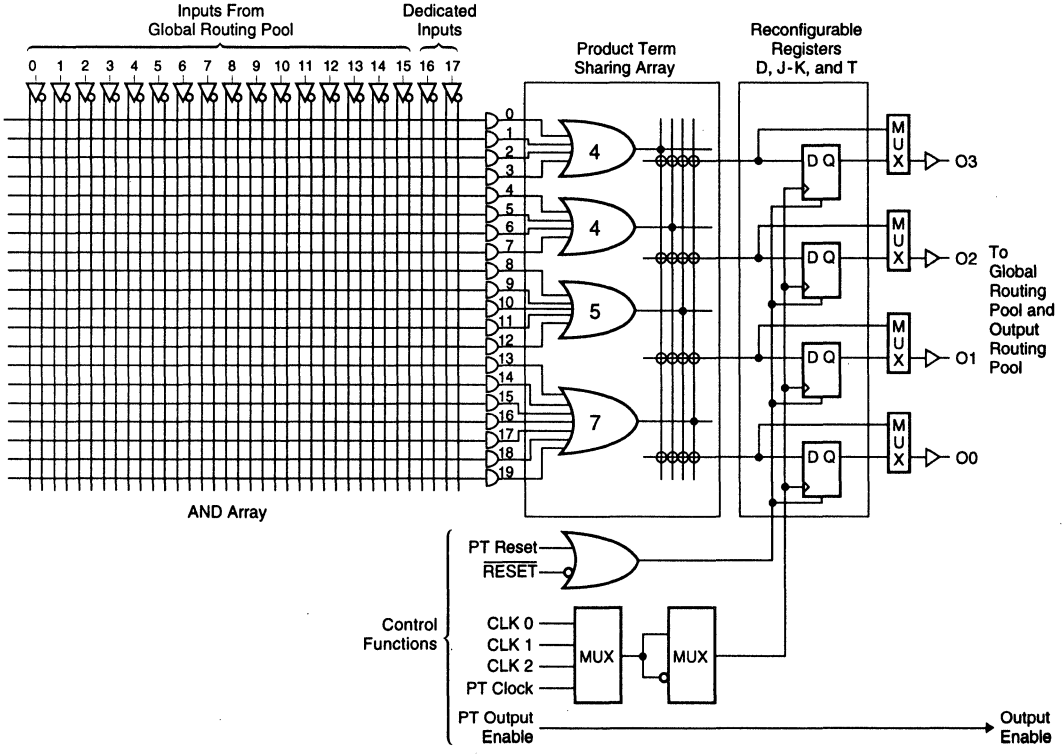
GLB Partitioning

The pDS software requires the designer to do manual partitioning and minimization of the design (the software does minimization within the GLB). This manual partitioning and minimization requires the designer to understand the pLSI architectural features and how to use them efficiently to do the best design.

The GLB features which the designer needs to understand and use properly to get the best performance from a design are shown in figure 7-1 and are as follows:

- Product Term Groupings
- Product Term Control Signals
- Product Term Sharing through the PTSA
- Hardware XOR
- 4 Product Term Bypass

Figure 7-1. GLB block diagram



GLB Partitioning

Refer to figure 7-6 for a quick reference chart showing the product term grouping for various options within the GLB. This chart will be used as the different features are discussed.

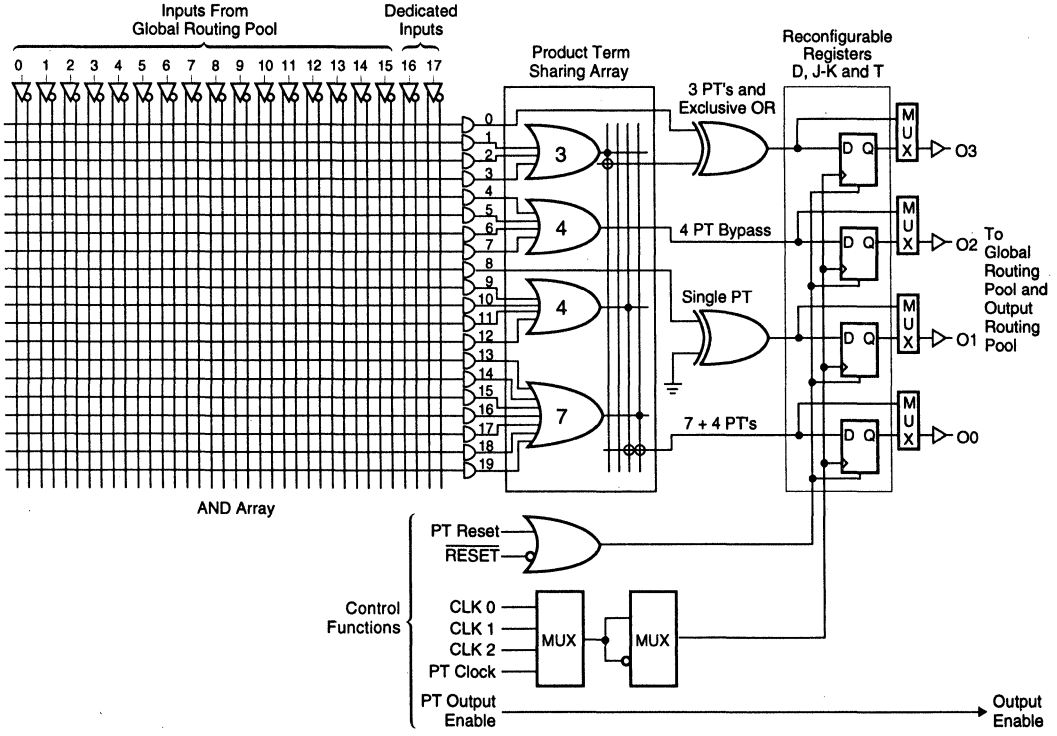
Product Term Groupings

The GLB has a default Product Term (PT) grouping of four, four, five, and seven PTs (please refer to section 4 of this data book for a detailed hardware description of the GLB). The designer needs to keep these groupings in mind at all times. The better the design can be fit into these default sized groups the better the chip utilization will be. To use these default groupings when the design is entered, the designer should not use GLB keywords. If keywords are not used, the software will automatically fit the logic into the default size groups. The basic size of each group does change, however, when other features of the GLB are used and these will be covered below (see figure 7-2).

Product Term Control Signals

As discussed in section 4 of this data book, there are two PTs which can be used to control different features in the GLB. These are PT12 and PT19 and they control the PT clock, the local GLB reset and the Output Enable (OE) from the GLB. As a reminder, only two of the three possible control signals can be generated in a single GLB. PT12 controls the PT clock or the GLB reset. PT19 is used to control the OE or the GLB reset. When the device is in its default four, four, five, seven configuration and when PT12 is used it will reduce the five PT group to four PTs, likewise when PT19 is used it will reduce the seven PT group to six PTs. Examples 1 and 2 are examples of the syntax used to invoke these features.

Figure 7-2. GLB Block Diagram Sharing PTSA Configured in Various Ways.



EXAMPLE 1: PTCLK & RESET usage

```
SIGTYPE A REG OUT;
EQUATIONS
   A.PTCLK = SIG1 & GATE;
   A.RE = DONE & !START;
   A = (P & D & !Q) # (I & Q);
END
```

In example 1 both PT12 and PT19 are used which causes the default PT groupings to become four, four, four, six. Example 2 shows the syntax used to get an OE from the GLB.

EXAMPLE 2: OE usage

```
SIGTYPE A OUT;
EQUATIONS
   A = W & X & !Y;
   B.OE = X & Y & Z;
END
```

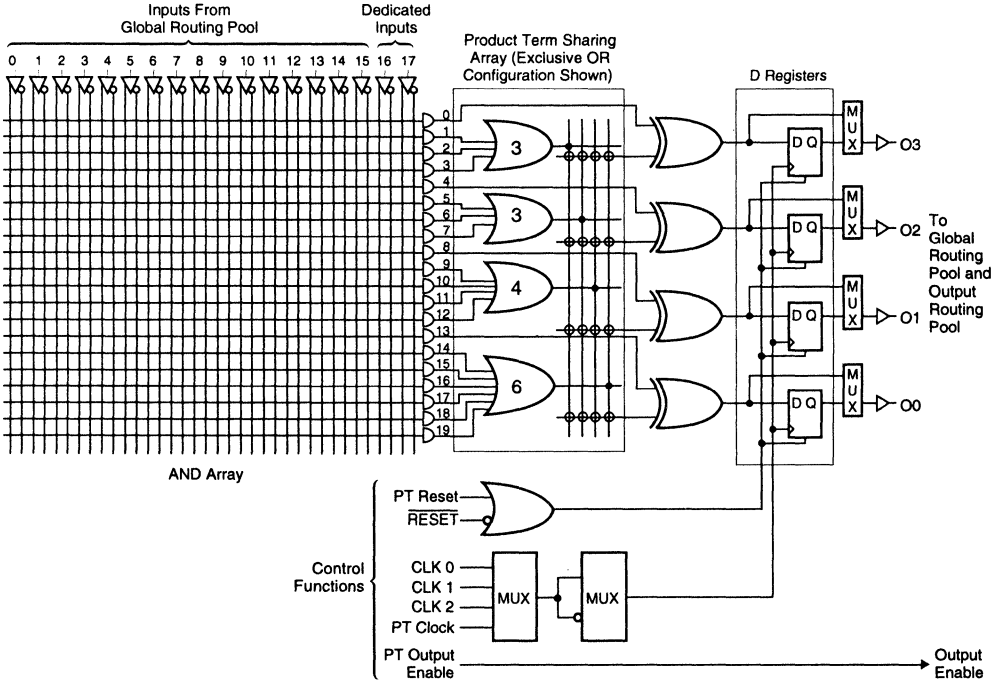
When an OE is specified in a GLB, PT19 is used, which reduces the seven PT group to six PTs. Designers must keep in mind that the PT group size changes when they use these control features.

The OE should not be set to V_{CC} or Gnd. This will use up PT19 and routing resources needlessly.

Product Term Sharing Array

The PTSA in each GLB provides a second level of OR-ing. This allows PT groups to be combined without an increase in device delay. This provides larger numbers of PTs for each equation, up to 20 PTs per equation. The basic size increments are the same as described above (four, four, five and seven), which equates to basic product term sizes of four, five, seven, eight (four + four), nine (four + five), etc.. Designers should keep these basic size increments in mind when dividing designs into different GLBs.

Figure 7-3. GLB with hardware XOR



Hardware XOR

The hardware XOR, which is included in the data path to each D-type register in the GLB, provides additional flexibility for the design implementation. It provides a means of implementing T-type and J-K flip-flop functions from the D-type flip-flop, as well as effectively adding an additional level of XOR in each GLB without needing to use feedback around the GLB. Figure 7-3 shows how the hardware XOR fits in the GLB.

In example 4, the hardware XOR is used to create a three input XOR. The hardware XOR is specified by using the $$, and the soft XOR, implemented as Sum-of-Product e.g. (C&!D) # (!C&D), is used with a $ symbol. The designer should feel free to use the hardware XOR, but should normally use the soft XOR to allow the software to use the hardware XOR as an inverter for possible minimization of PTs used.

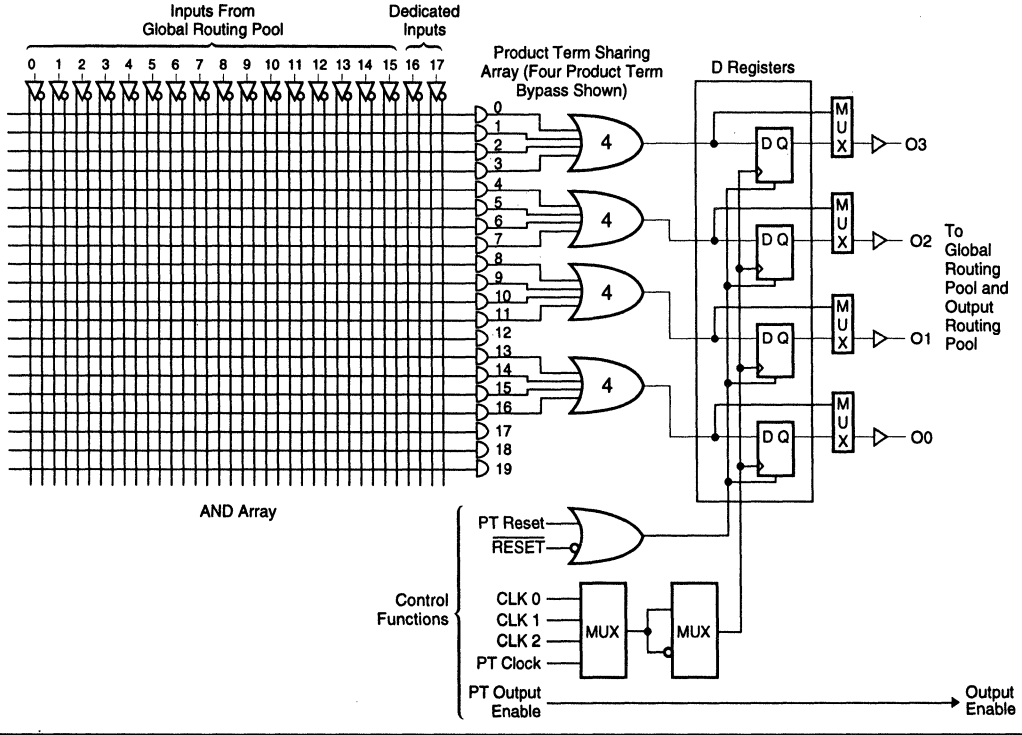
EXAMPLE 4: Hardware XOR usage

```
SIGTYPE A REG OUT;
EQUATIONS
   A.PTCLK = SIG1;
   A = B $$ (C $ D);
END
```

Product Term Bypass

The GLB offers a data path which bypasses the PTSA and hardware XOR. This path is referred to as the four PT Bypass, and provides a faster signal path for equations of four or less PTs. Each output of the GLB can use the four PT bypass on an individual basis by assigning the keyword CRITICAL to the SIGTYPE statement for that OLMC. This feature should be used when high speed performance is required and the equations need four or less PTs. Figure 7-4 shows all four OLMCs configured in 4 PT bypass mode.

Figure 7-4. GLB with 4 Product Term Bypass.



In Example 5 the keyword CRITICAL is used to make the "A" signal path use the 4 PT bypass.

EXAMPLE 5: CRITICAL usage

```
SIGTYPE A CRITICAL OUT;
EQUATIONS
   A = B # C # D;
END
```

The Fitter

In the verification step of the design process, the pDS software system provides a Fitter function which performs the following operations on the design which is being entered;

- Minimizes the equations to reduce the number of Product Terms used to implement the logic.
- Maps the equations into the device Product Terms.
- Generates Output Duplication and Output Swapping options for the router (see the section on Understanding Router Implications for more details).

Minimization

The software provides a logic minimization function which works to reduce the logic entered into the minimum number of product terms possible. This feature is needed to make better use of the resources available in the GLB. Another benefit of minimizing the number of Product Terms needed in the design is that the design can often be implemented in fewer levels of logic providing a faster design.

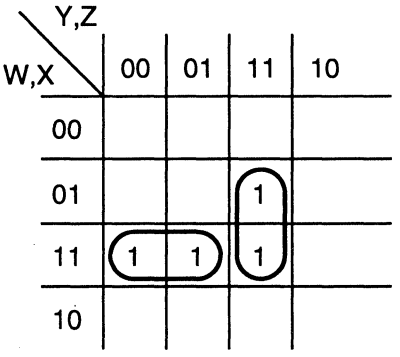
There are times when the designer will not want the logic to be minimized. An example exists with the possibility of Hazard states in the design and extra product terms are needed to eliminate them. As shown in example 6, the original equation input can be minimized from three Product Terms to two Product Terms.

EXAMPLE 6: Minimized Equation

ORIGINAL	MINIMIZED
A = W & X & !Y	A = W & X & !Y
# W & X & Z	# W & Y & Z;
# X & Y & Z;	

Logically these two equations will function the same, however, looking at the Karnaugh map, (figure 7-5) the third Product Term is needed to eliminate the Logic Hazard state between the two minimized Product Terms which can cause glitches.

Figure 7-5. Logic Karnaugh map



The software provides a Keyword in the syntax which turns off the minimizer for individual equations. This Keyword is NOMIN and is shown in the following example.

EXAMPLE 7: NOMIN usage

```
SIGTYPE A NOMIN OUT;
EQUATIONS
  A = W & X & !Y
    # W & X & Z
    # X & Y & Z;
END
```

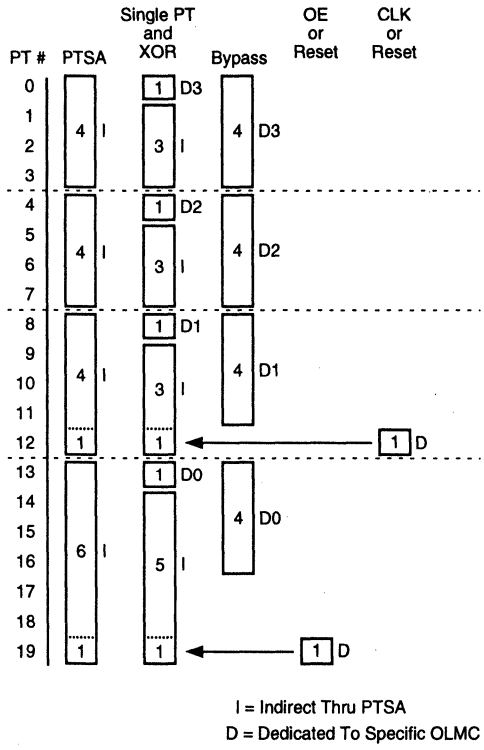
Router Support

After the fitter has performed minimization and mapping in the GLB, it generates tables for the router. These tables allow the router to do Output Duplication and Output Swapping. These two functions are discussed in detail in the Router section. Designers can help design partitioning, if the Router section suggestions are followed. The designer is responsible for partitioning the design in the GLBs and there can be many possible combinations of how the logic is split and fit into the GLBs. Designers must recognize the different options available, and use the appropriate one to satisfy the design constraints.

Fit Examples

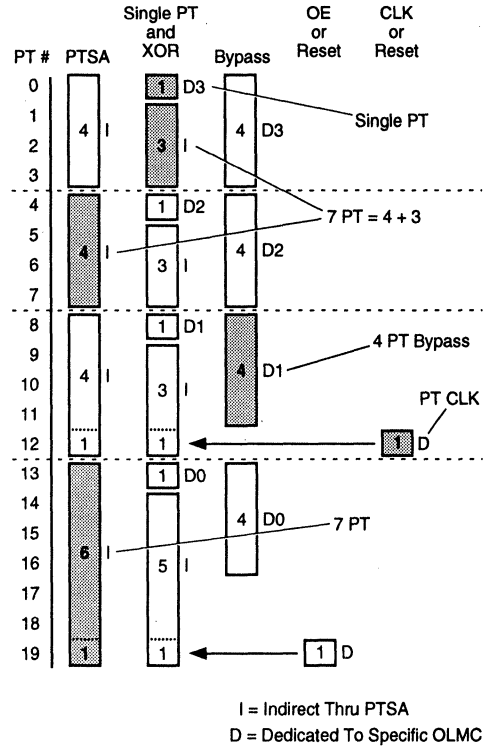
Figure 7-6 provides a Product Term sharing chart which will be used along with some examples to help clarify the PT groupings and how to use the chart to determine logic fit within the GLB.

Figure 7-6. Product Term Sharing Chart



The first example (see figure 7-7) shows a proposed fit with four PTs used in a Critical path and PT groups of one, seven, and seven needed. A PT clock is also needed in this GLB which gives a total of 20 PTs required. The gray boxes indicate the way the PTs were fit into the GLB and that this fit will work.

Figure 7-7. Fit Example #1



The second example (see figure 7-8) shows a proposed fit with PT groups of one, two, seven, and seven needed. A PT clock as well as an OE is generated in this GLB which gives a total of 19 PTs needed. The gray boxes indicate the way the PTs were fit into the GLB and that this fit will not work. This proposed fit will work if the OE signal is eliminated from this GLB and moved to another GLB in the Megablock.

The third example (see figure 7-9) shows a proposed fit with the PT groups of two, two, two, and seven needed. An OE is also needed in this GLB which gives a total of 14 PTs required. The gray boxes indicate the way the PTs were fit into the GLB and that this fit will not work. It should be noted that the groups of two PTs use up 13 PTs for only six needed. The OE in this example is again the signal which would be the first candidate to be moved to another GLB.

From these three examples, it becomes apparent that designers must consider the total number of PTs needed per GLB as well as the grouping of the PTs.

Figure 7-8. Fit Example #2

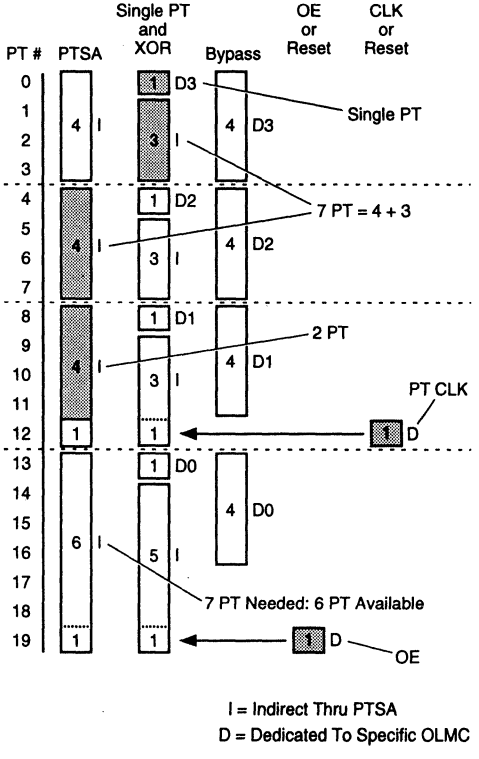
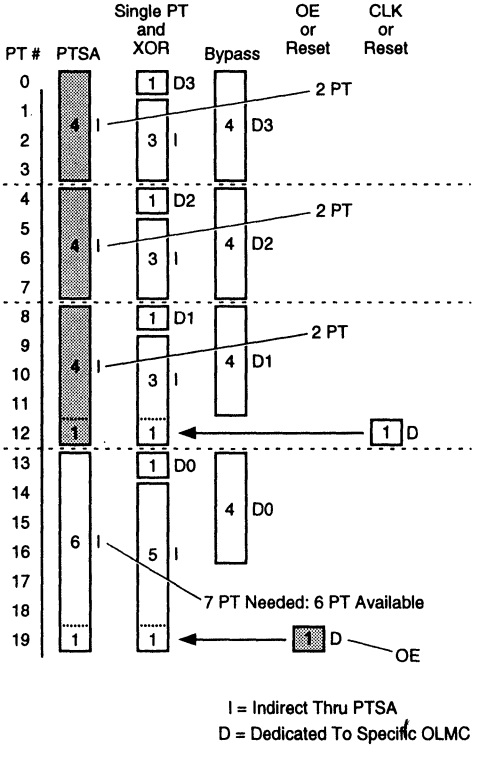


Figure 7-9. Fit Example #3



7

Design Optimization: Placement and Routing

Introduction

The architecture of pLSI and ispLSI devices is optimized to offer maximum speed as well as flexibility of design. The interconnects for both pLSI and ispLSI devices have been optimized for fast performance and predictability of speed. They offer complete connectivity from any I/O pin input to all of the GLBs of the device and similarly from any GLB output to the inputs of all of the GLBs. pLSI and ispLSI devices use a proprietary, intelligent interconnect scheme called the Global Routing Pool (GRP) to provide both performance and connectivity. Along with the GRP is another routing resource called the Output Routing Pool (ORP), which provides flexibility in connecting various GLB outputs to the I/O pins. The two interconnects require an intelligent Logic Placement and Routing (LPR) software algorithm which offers designers fully automatic routing of their designs. Lattice offers the LPR program as an integral part of its pDS software. This section offers an insight into the LPR and provides some helpful tips to maximize the routability of designs. It covers:

- Determining Routability
- Interconnect Structure & LPR
- Input Congestion
- Rules for Fixing/Assigning Pins
- Using Design Features/Options

Determining Routability

Prior to discussing the details of LPR, some general guidelines are provided. With utilizations of up to 80% of the device, Lattice assures 100% routability of designs. The utilization density is a measure of the number of GLBs used in the device. To determine a design's utilization, count the number of GLBs used within a design and ratio it to the total number of GLBs available in the device. A design using fewer than 80% of available GLBs will probably be successful in routing without any user intervention.

Fixing signals to specific I/O pins or input pins acts as a constraint to the LPR and tends to reduce the routability of a design. Similarly, using certain device features such as output enables, output bypass, dedicated input pins and the clock GLB tends to reduce the routability of a design. The user should generally minimize the usage of these features and at least ensure that the first routing of the design is done with all pins free to maximize the routability of the design. A detailed description of the implications of fixed pin assignments and usage of these other GLB features is provided in this section.

The LPR software sometimes resorts to using such contention breaking schemes as splitting a GLB logic into two different GLBs or duplicating a GLB output signal. To allow the LPR software to do the splitting and duplication, designers should leave a few empty GLBs and a few unused outputs in the GLBs. This will ensure that a spare GLB or GLB output is available to the LPR software to resolve a contention.

When a design becomes unroutable, it can be due to any of the above reasons. The designer should carefully follow recommendations in the subsequent sections to ensure easy routability of designs. An important factor to remember is to reduce the device utilization by splitting logic in GLBs with the highest number of inputs. This tends to reduce GRP congestion most and will help improve routability with minimum loss of utilization.

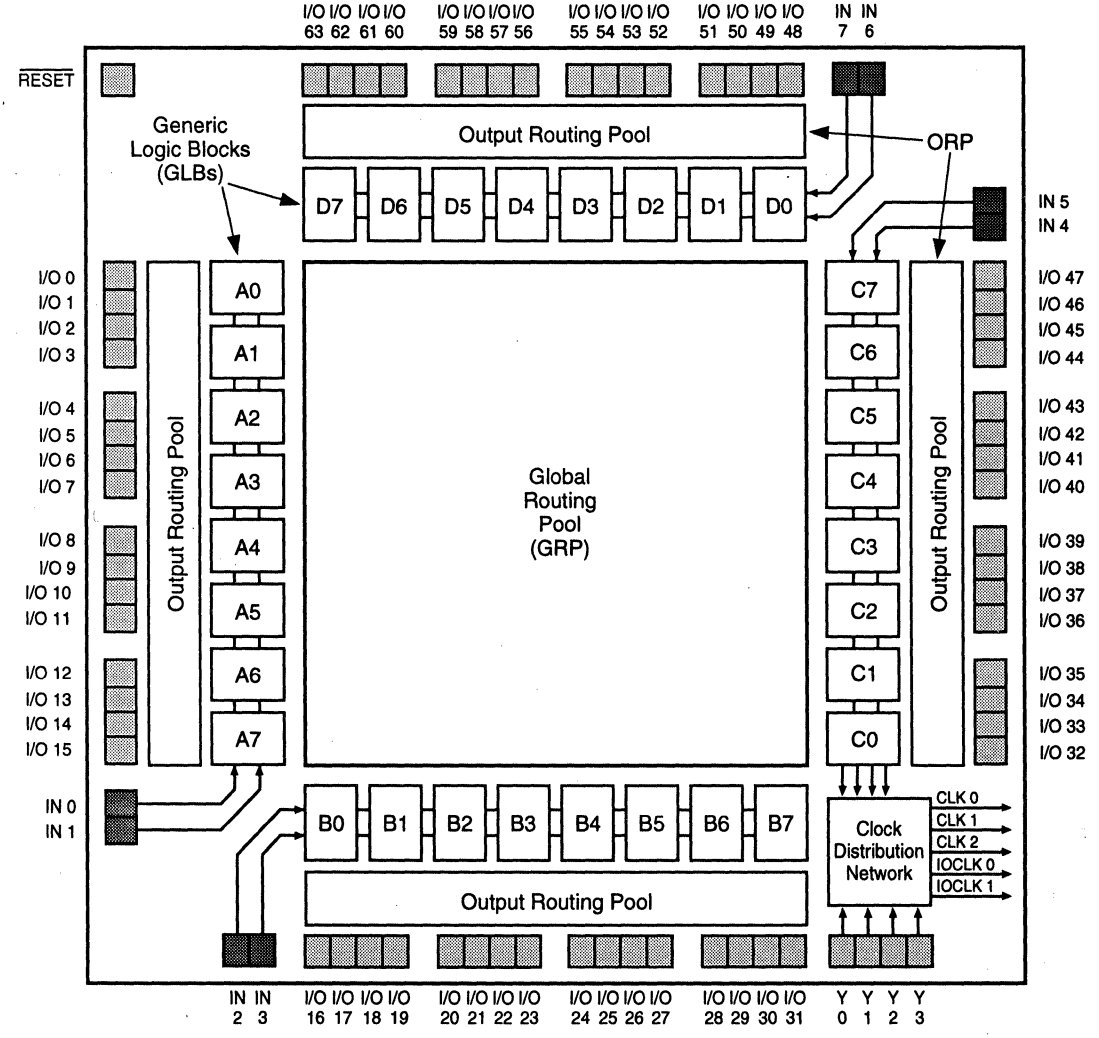
Interconnect Structure & LPR

pLSI and ispLSI devices use the GRP as their main routing resource (see figure 7-10). Through the GRP, all of the GLB output signals and I/O pin input signals can be routed to all other GLBs in the device. These signals are available to GLBs on certain specific GLB input lines only. The selection of appropriate GLB inputs is done automatically by the LPR software. Details of the GRP architecture are explained in section 4 of this databook.

The LPR software moves logic to different GLBs within the device to decide on an optimal placement, which allows all GLB outputs to be connected to the appropriate GLB inputs in the device. This is required since GLB outputs are available to other GLBs on certain input lines only. It also swaps the four GLB outputs to decide on the optimal placement, overcoming the same restriction. Similarly, the LPR moves various I/O pin input signals to ensure that they are available to all appropriate GLBs which require the input signal. Using the GRP, the LPR uses the following techniques to route a design:

- Move Logic to Various GLBs
- Swap the Outputs of GLBs
- Move Input Signals to Various I/O Pins
- Duplicates Logic Inside GLBs

Figure 7-10. GRP is the Main Routing Resource



The outputs from a group of eight GLBs are routed to 16 I/Os using an ORP. This group of eight GLBs and 16 I/O pins is called a Megablock (see section 4 for details). Any GLB output in a Megablock can be routed to four different I/O pins associated with the Megablock, providing flexibility in pin assignment/fixing (see figure 7-11). Unless it is user defined, the selection of the appropriate I/O pin for output is also done automatically by the LPR software. The LPR first places the GLB with the output signals within the Megablock and then it uses this freedom to select any of the four possible I/O pins as outputs. If required, it also swaps the GLB signals to allow connection to appropriate I/O pins. Using the ORP, the LPR software uses the following techniques to enhance the routability of the design.

- Move Logic to an Appropriate GLB within a Megablock
- Swap the Outputs of a GLB
- Move Output Signals to Various I/O Pins within a Megablock

Input Congestion

Fundamentally, all of the techniques used by the LPR are to route various signals to GLB inputs. If the routing fails, it is due to the input congestion in various GLBs. The LPR software uses two techniques to enhance the routability of designs. Both these techniques are transparent to the designer.

- GLB Splitting
- Duplication of GLB Outputs

Splitting of GLBs into two different GLBs has two benefits. First, it allows the LPR to find additional routing channels to allow source GLB signals to connect to the inputs of the destination GLBs. Second, it frequently reduces the number of inputs of the split GLBs, reducing input congestion. It is a good design practice to leave a few unused GLBs in the design, allowing the LPR software to use these for GLB splitting. This will enhance the routability of designs, however it is not uncommon to find designs with all GLBs used.

Duplication of GLB output signals is another technique used by the LPR to find additional channels for source GLB signals to reach destination GLB inputs. To allow the LPR to duplicate GLB signals, it is a good practice to leave some GLB outputs unused. This facilitates better routability of designs.

The duplication of GLB outputs has design implications, especially for registered outputs with asynchronous inputs. Figure 7-12 shows the same registered signal duplicated on two different GLB outputs. These two registered outputs are supposed to have identical logic during operation. However, the asynchronous input signal SIG 1, due to different delays to the two registers, may set different logic levels in the two registers which could cause a system failure. To avoid such a situation the pDS software offers the ASYNC flag which prohibits the LPR from duplicating the signal. The syntax used for the ASYNC flag is shown in example 8.

EXAMPLE 8: Syntax for using ASYNC flag in pDS software

```
SIGTYPE SIG_1 Async Reg Out;
Equations;
    SIG_1=CTL3 & DATA_SIGNAL;
     SIG_1.clk=system_clock1;
END;
```

Excessive use of ASYNC flag tends to restrict the ability of LPR software to duplicate signals, consequently reducing routability of designs. Designers must use the ASYNC flag prudently, where required.

Finally, since GLB inputs are the main constraints, the routability of designs can be enhanced by reducing the number of GLB inputs. In general the GLBs with the highest number of inputs will be the toughest to route. The designer can manually split a few of these GLBs to improve routability of the design.

The general guidelines for ensuring successful routing which are based on input congestion are:

- Always leave a few GLBs unused
- Always leave a few GLB outputs unused
- For asynchronous input signals use ASYNC flag prudently
- Split a few GLBs which have a large number of inputs

Figure 7-11. ORP Routes GLB Outputs to I/O Pins

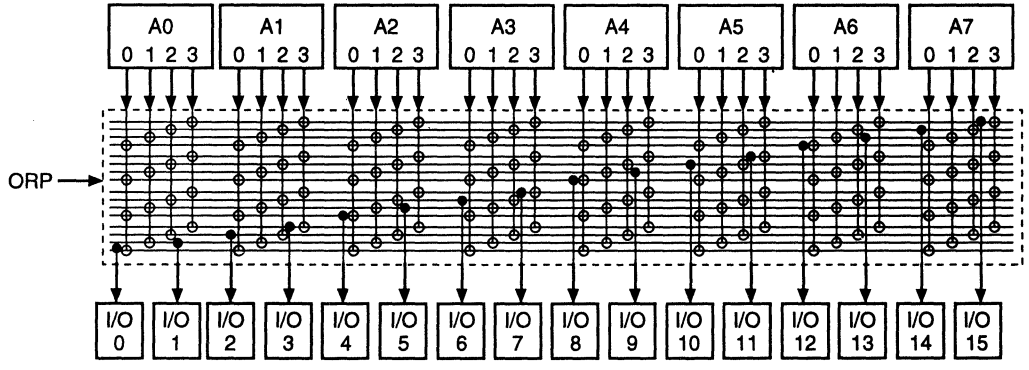
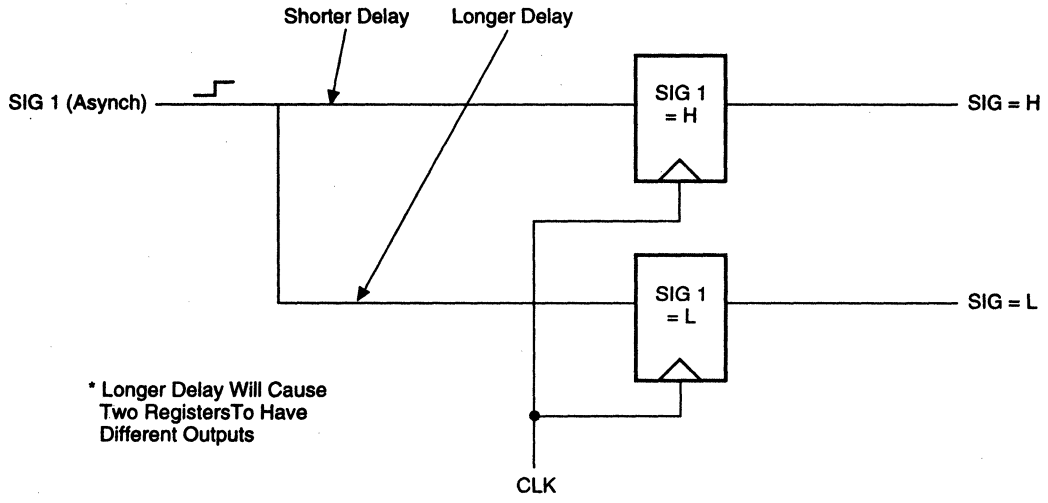


Figure 7-12. Asynchronous Inputs to Duplicated Signals



Rules for Fixing/Assigning Pins

As described in the section on LPR, the freedom of moving input and output pins is important from the GRP and ORP standpoint, respectively. The LPR uses this freedom to ensure routability. Conversely, fixed pins are frequently required to meet design or board constraints. The pDS software system allows the user to fix signals to specific pins using the keyword LOCK. The syntax for fixed pin assignment is shown in example 9.

EXAMPLE 9: Syntax for Fixing signal to pin

```
XPIN IO REG_OUT LOCK 12;
OB11 (REG_OUT, REGISTER_OUT);
```

Limiting the number of fixed pins helps enhance routability of a design. Whenever possible, the first routing of the design should be done without fixed pins. Subsequent routing should attempt to reduce the number of fixed pins in a design. A similar issue relates to GLB architecture. The LPR uses GLB output swapping to find an appropriate input for the destination GLB.

Fixing of I/O pins as inputs imposes one more restriction for a design due to the GRP structure. No two I/O pin inputs to the device which go to the same GLB can be placed 16 I/O pins apart. This is because the GRP allows device I/O pin inputs to specific GLB inputs only.

Similar to the GRP, the ORP places two more restrictions on the LPR for fixing pins. First, it restricts the placement of the GLB with the fixed output signal to a specific Megablock. In other words an output from a GLB cannot go to I/O pins in different Megablocks (see figure 7-10). The LPR has the capability to split such GLBs into two different GLBs, at the cost of device utilization. Second, since for every GLB output the ORP offers a choice of four I/O pins within the same Megablock, fixing outputs from the same GLB to I/O pins four apart makes the design unroutable (see figure 7-13).

The rules to follow are:

- Whenever possible, the first route should be with no fixed pins
- Limit the number of fixed pins for subsequent routes
- Never fix I/O pin inputs to same GLB, multiple of 16 apart

- Preferably fix outputs from the same GLB to I/O pins within a Megablock
- Never fix outputs from a GLB in multiples of four I/O pins apart

Using Design Features/Options

Key considerations for the LPR are: moving logic amongst various GLBs, moving inputs and outputs amongst I/O pins and swapping GLB outputs. There are many features in pLSI and ispLSI devices which offer additional functionality; however, there can be adverse affects with these considerations. Principal features to consider while routing a design are:

- ❑ Bypass Options for Fast Speed
- ❑ Dedicated Inputs
- ❑ Output Enable
- ❑ Clock GLB

Bypass Options for Fast Speed

As described in the section on GLB Partitioning, the Product Term Sharing Array (PTSA) is used to swap the GLB outputs. The GLB four product term bypass option offers faster speed, but does not use the PTSA. These four product term bypass outputs are defined by the designer using keyword CRITICAL in the GLB definition. While they offer faster speeds, the four product term bypass outputs restrict the flexibility of GLB outputs. This reduces the ability of the LPR to find an appropriate routing channel to the destination GLB input, consequently effecting routability. The fast outputs should be used only when necessary to enhance the performance of designs.

The ORP in pLSI and ispLSI devices offers another capability. It allows users to bypass the ORP using a fast bypass option shown in figure 7-14. This bypass option is selected by the user using keyword CRITICAL in the I/O cell definition as shown in example 10. Only two outputs per GLB can be bypassed to I/O pins. LPR does not split due to output bypass, it ignores output bypass if required for routing. Furthermore, it may use bypass on non-CRITICAL output if needed for routability.

EXAMPLE 10: Syntax for ORP bypass defined in I/O Cell

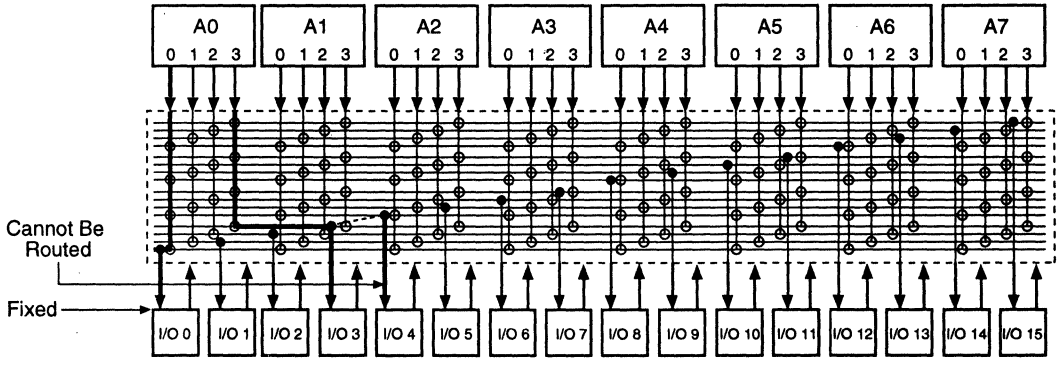
```
XPIN IO REG_OUT CRITICAL LOCK 12;
OB11 (REG_OUT, REGISTER_OUT);
END
```

Based on this discussion, the rules to follow are:

- Prudent use of four product term bypass options
- Wherever possible define only two ORP bypass options for a single GLBs outputs
- Prudent use of ORP bypass

7

Figure 7-13. Output From Same GLB Cannot be Four I/O Pins Apart



Dedicated Inputs

The pLSI and ispLSI devices offer a number of combinatorial dedicated inputs (see figure 7-10). Two dedicated inputs are routable to only the eight GLBs in any given Megablock. Unlike I/O pin inputs which can be routed to all of the GLBs in the device, the dedicated inputs are routable to only eight GLBs. Locking signals to dedicated input pins reduces the ability of the LPR to move logic between various GLBs. Whenever possible, the dedicated input pins should not be fixed. Also, since dedicated inputs are routable to at most eight GLBs, no input signal which is used in four or more GLBs should be fixed to dedicated input pins to enhance routability.

While it may appear that dedicated input pins are appropriate to route common control signals to multiple GLBs, in reality they should carry signals required by the fewest number of GLBs. For example a loadable counter design requires both Count Enable and Load Data signals. The Count Enable signal is required by multiple counter bits, whereas, the Load Data signal is required by only one count bit. It would be prudent to use the dedicated input pins for Load Data signal, instead of Count Enable signal since this would allow maximum movement of logic among various GLBs. The LPR software automatically assigns a low fanout signal to the dedicated input pins and moves a higher fanout signal to an I/O pin. When using dedicated inputs, the rules to follow are:

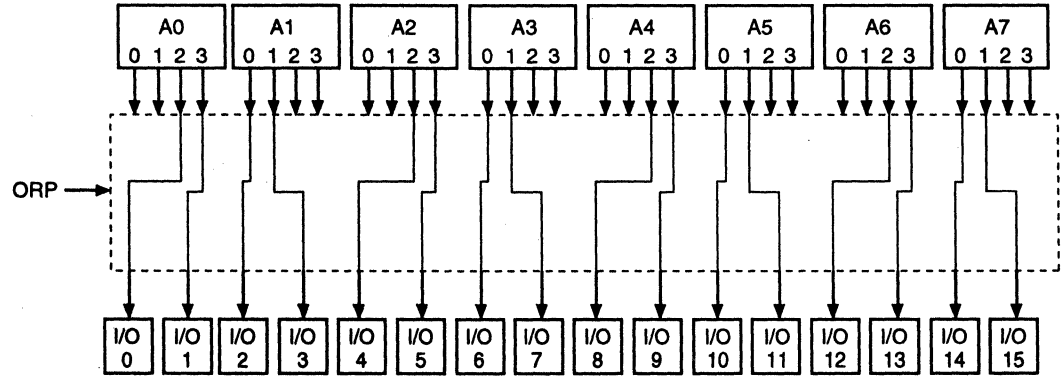
- Whenever possible, do not fix dedicated inputs
- Never assign signals required by more than 4 GLBs to dedicated inputs
- Use low fanout signals for dedicated inputs

Output Enable

For 16 I/O pins in every Megablock there is one output enable signal. This signal can be generated by a product term in any of the eight GLBs of the Megablock (see figure 7-15). All I/O signals controlled by the same output enable signal are automatically grouped by the LPR into the same Megablock. Fixing I/O signals into two different Megablocks will make the design unroutable. Similarly fixing I/O signals controlled by different output enables in the same Megablock will also make designs unroutable. Generating the output enable signal in the same GLB where the output signals are generated enhances routability of designs. Placing I/O signals using different Output Enables in the same GLB will make designs unroutable. Sometimes moving the output enable product term to a different GLB will also help route a design. When using output enables, the rules to follow are:

- Never fix I/O signals using the same output enables in different Megablocks
- Never fix I/O signals using different output enables in the same GLB or Megablock
- Moving the output enable signal to a free GLB enhances routability

Figure 7-14. The ORP Bypass Options



Clock GLB

Both pLSI and ispLSI devices offer the capability of generating internal synchronous global clocks using the Clock Distribution Network. These clocks are generated in a single GLB dedicated for this purpose (GLB C0 in pLSI 1032). The clock GLB can also be used as a logic GLB, i.e. if the outputs of clock GLB are not used for generating internal synchronous clocks, they can be used for logic functions. It is feasible, for example, to use two of the clock GLB outputs as clocks, while the other two are used as logic signals. Since the location of the clock GLB is fixed, the logic defined in the clock GLB cannot be moved to any other GLB. This restricts the ability of the LPR to move logic amongst various GLBs as well as the ability to swap clock GLB outputs. Mixing logic and clock into the clock GLB will therefore adversely effect the routability of designs. When using the clock GLB, the designer should attempt to use it for only clock generation or only logic functions.

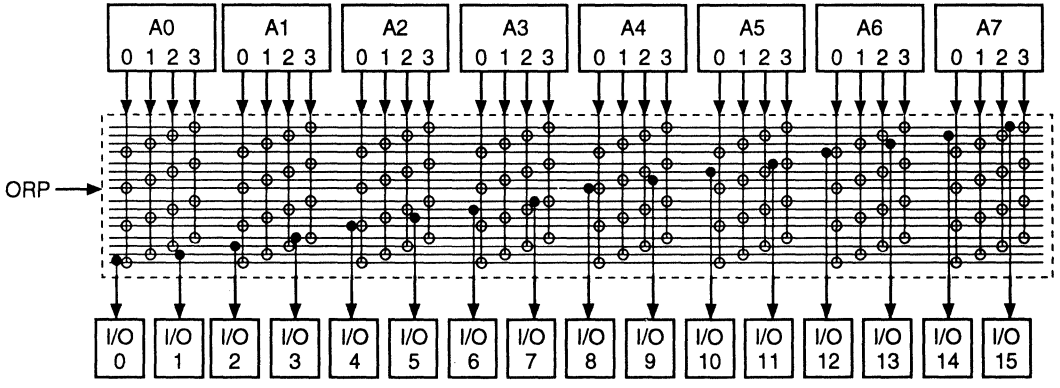
The rule to follow is:

- Whenever possible, do not mix clock generation and logic functions in the clock GLB

Conclusion

We have examined in detail various aspects affecting the routability of designs for the pLSI and ispLSI family of devices. While it is important to understand these considerations, most designs with 80% utilization should 100 % route without any user intervention. In fact designs with much higher utilization will also route easily. The 80% utilization figure is really a design guideline for using these devices.

Figure 7-15. Output Enable within a Megablock



Design Optimization: Device Utilization

Introduction

Design techniques for performance and speed have been discussed in previous sections. Another important consideration is designing for utilization. General guidelines and examples on how to design for better utilization are highlighted in this section.

Utilization Calculation

Lattice recommends that for 100% routability, the utilization be kept to 80%. Utilization is defined as a percentage. To calculate utilization, the total number of used GLBs in a design are divided by the total number GLBs available (see example 11). The 80% utilization is based on the fact that 20% of the GLBs in the device are totally unused.

Architecture and Utilization

It is important to remember the architecture of the pLSI and ispLSI devices when designing and implementing logic. One of the key features of these families is the 18 inputs per GLB. Using the wide gating of the GLB, rather than cascading logic, improves both utilization and performance by reducing levels of delay. An example of this type of logic design is a counter.

In figure 7-16, there are two 8-bit counters. To create a 16-bit counter, a carry out must be implemented which uses valuable resources. A better way to build this function is to build a 16-bit counter and take advantage of the wide gating. This reduces the levels of delay, and more importantly, frees unnecessary GLB outputs for other uses.

State Machines and Utilization

Another way to maintain high utilization is to create state machines instead of counters with separate decoders. This is related to logic implementation. Figure 7-17 and figure 7-18 show this example in a block diagram format.

Figure 7-17 shows a 4-bit counter with logic to detect or decode the terminal count. The logic uses 1.5 GLBs including a carry out. A seven segment decoder (binary to seven-segment) uses 1.75 GLBs. The logic uses a total of 3.25 GLBs. The logic can be reduced to 3 GLBs, if the carry out is eliminated from the counter.

The same function is accomplished by designing a 7-state machine; one state for each segment. This design utilizes only 1.75 GLBs and if the Carry Out is needed, the design adds only 0.25 GLBs. This is a significant savings in utilization, from 3.25 GLBs down to 2 GLBs. Figure 7-18 shows the seven state machines without the carry out. The state logic is designed to decode only the correct signal necessary to activate the particular segment it is meant to enable. The numbers inside of the *OR* gates indicate the number of product terms needed for that particular segment. The partitioning of the logic is user-definable.

EXAMPLE 11: Calculating Utilization

$$\frac{\text{Number of GLBs Used}}{\text{Total Number of GLBs Available}} \times 100 = \text{Utilization \%}$$

Figure 7-16. Counters

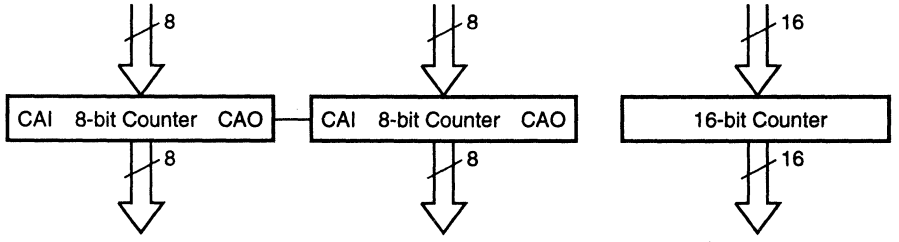


Figure 7-17. Counter and Seven-Segment Decoder

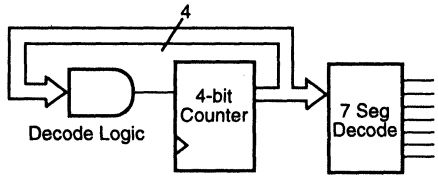
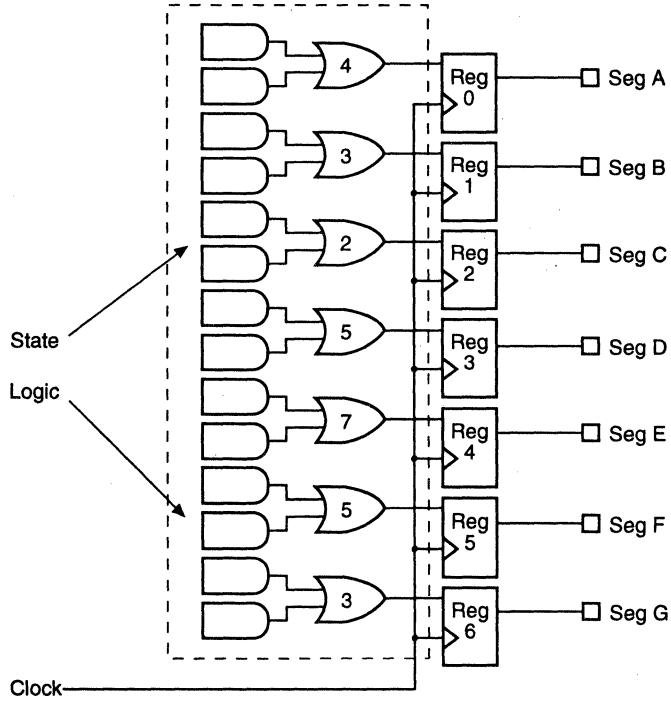


Figure 7-18. Seven-Segment State Machine



Comparators can also be implemented in several configurations. Figure 7-19 shows a cascaded implementation of a 2-bit comparator. This implementation requires 0.75 GLBs, six product terms and three GLB outputs.

Figure 7-20 shows a parallel implementation which only requires 0.25 GLBs. This implementation uses more product terms, because the true and complement for each input of the comparator are required. The maximum comparator that can be implemented in a GLB is 8-bits. This is also a significant decrease in utilization compared to a cascaded version. As mentioned above, the parallel version of an 8-bit comparator uses 16 product terms and only one output of a GLB, while a cascaded 8-bit comparator requires 2.75 GLBs and three extra levels of delay. The combination of the cascaded and parallel implementation are ideal for larger comparators.

Figure 7-19. Cascaded 2-bit Comparator

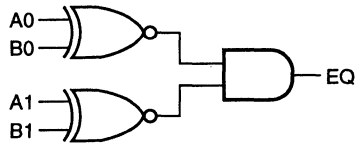
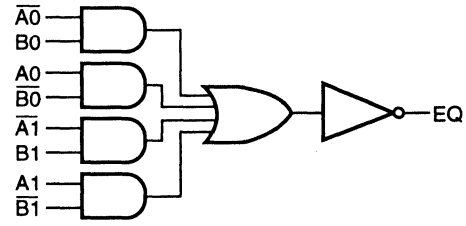


Figure 7-20. Parallel Two-bit Comparator



Architectural Features

Other architectural features to consider when partitioning your logic are:

- Product Term Sharing Array (PTSA)
- Output Enables (OE)
- Global Routing Pool (GRP)

Product Term Sharing Array

The PTSA redistributes the Product Terms (PT) in other configurations besides the default (four, four, five, seven). Using PTSA, OR functions with more than seven PTs within a GLB can be generated. This is accomplished by combining OR Product Terms. The PTSA also allows logic within a GLB to share OR terms, if the logic is common to two or more PTs. This minimizes logic and increases utilization.

Output Enable

The Output Enable (OE) signal is generated from any GLB within a Megablock. Because the OE signal is generated on a specific PT, the logic partitioning can reduce utilization. If an equation requires an OE and seven product terms in the same GLB, the logic cannot fit due to the OE requiring one of the seven PTs. By moving the OE equation to another GLB that is not using the OE product term, utilization is increased.

Global Routing Pool

The GRP can also increase utilization. The GRP connects either the true or the complement of any signal to any GLB. If the complement of a signal is required within a GLB, use the complement as an input to the GLB rather than generating a separate complement signal using another GLB output.

Hard Macros

The use of hard Macros increases utilization. Hard Macros are configured to be efficient and use the least amount of resources. Some functions are performed with hard or soft Macros. Whenever possible, you should elect to use the hard Macro version.

In summary, you should compress logic into an AND/OR format on a GLB basis and take advantage of the architecture, such as the wide gating, when implementing logic. Some of the techniques mentioned also increase circuit speed.

Design Optimization: Speed

Introduction

The designer is often faced with design requirements calling for approaches which will get the most speed from a design. Often getting the fastest performance will sacrifice some device utilization. It is up to the designer to make the final trade-off between device utilization and speed of operation.

General Guidelines

As the designer approaches the design, there are a few general rules which should be followed for best speed performance, as listed below;

- ❑ Always reduce the design to the fewest levels of logic possible.
- ❑ Take advantage of the faster paths through the device whenever possible.
- ❑ Implement synchronous designs when possible.
- ❑ Pipeline the design when possible to achieve the fewest equivalent levels of logic.

Device Timing

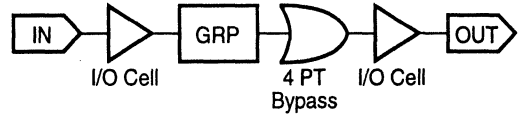
In general, logic designs can be divided into two categories, Direct Combinatorial and Synchronous (or registered). Most decoders, adders, etc. are examples of direct combinatorial (combinatorial), and familiar functions such as counters, state machines, etc. are examples of synchronous combinatorial (registered). Design approaches in terms of solutions for each categories will be discussed.

Combinatorial Design

As a starting point, a review of the 15 ns data path, as well as how it relates to combinatorial designs will follow. Referring back to the timing model discussion in Section 4 will help provide information on all the timing paths. The timing paths will be discussed here as they are used.

The 15 ns data path (for pLSI and ispLSI 1032-80) is achieved with a signal coming in on a general I/O pin (t_{din} = 2 ns), through the GRP (t_{grp4} = 3 ns), using 4 PT Bypass in the GLB (t_{4pt} = 6 ns), then out to an I/O pin (t_{ob} = 4 ns) as shown in figure 7-21. It should be noted that 4 loads on the GRP means the signal is driving 4 GLBs, not 4 PTs. 4 GLBs can be up to 80 PTs and 16 outputs.

Figure 7-21. Standard 15 ns Data Path



The following is a simple example to review some of the points the designer needs to consider to achieve the best speed/delay performance from the design.

EXAMPLE 12: Combinatorial

This example is a simple 4-bit comparator as shown in figure 7-22.

1. To get the best speed from a design the designer should try to always reduce the number of GLBs the logic signals go through.
2. When possible the designer should try to use the 4 PT bypass path within the GLB.
3. For the fastest output path the designer should use the ORP bypass.

Figure 7-22. Bit Comparator Schematic

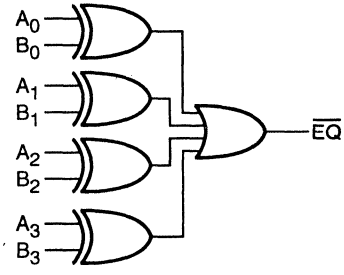
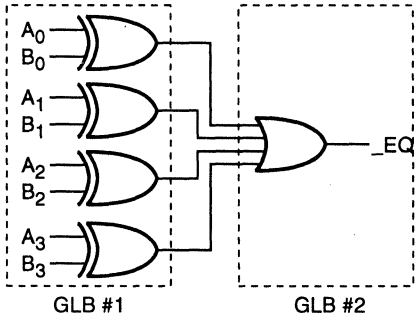


Figure 7-23 shows a valid way of implementing the Comparator which is logically correct. Listing 1 provides a copy of the LDF fragment which would implement this design.

Figure 7-23. The 4-bit Comparator Implementation



Listing 1. The 4-bit Comparator Implementation

```
// GLB #1
Equations
  E0 = A0 $$ B0;
  E1 = A1 $$ B1;
  E2 = A2 $$ B2;
  E3 = A3 $$ B3;
End
// GLB #2
Equations
  _EQ = E0 # E1 # E2 # E3;
End
```

A review of this implementation, will show that on the surface it looks good, but there are several things that can be done to improve the delay of the design.

This design implementation will have a delay of 27.5 ns, based on the delay values given in the pLSI 1032-80 Data Sheet (Refer to Section 2). This implementation is 10 ns slower than it needs to be, for the following reason:

> The design is partitioned so it uses two levels of logic. For this example it adds 4 ns for the added GRP delay (t_{grp4}) and 6 ns for the second GLB delay (t_{4pt}).

In example 13, the same function is implemented using a single GLB level which will give a delay of 17.5 ns based on the delay values given in the pLSI 1032-80 Data Sheet.

Example 13:

This example is an improved 4-bit Comparator which only requires a single GLB level of logic. Shown in listing 2, are the equations from listing 1 rewritten in a format which will minimize to the form shown in listing 3. The hardware XORs ($$) from the original equations were changed to soft XORs ($), which then allows the equations to be implemented in their sum-of-products equivalent. Figure 7-24 shows the schematic of this minimum implementation which requires 17.5 ns.

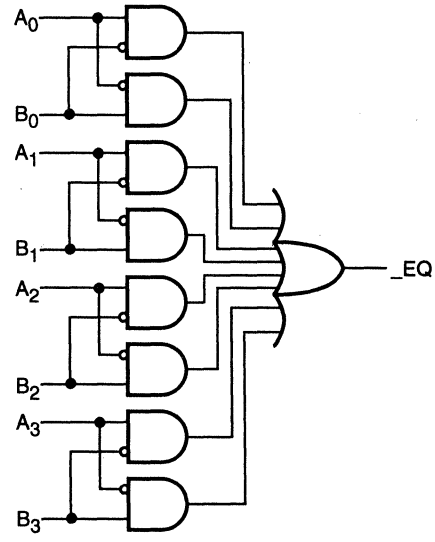
Listing 2. A 4-bit Comparator Using One GLB Level

```
Equations
  _EQ = (A0 $ B0)
      # (A1 $ B1)
      # (A2 $ B2)
      # (A3 $ B3);
End
```

Listing 3. Minimized Version of the Comparator

```
Equations
  _EQ = (A0 & !B0 # !A0 & B0)
      # (A1 & !B1 # !A1 & B1)
      # (A2 & !B2 # !A2 & B2)
      # (A3 & !B3 # !A3 & B3);
End
```

Figure 7-24. Improved 4-bit Comparator Schematic



The examples given show in a simple way how important it is to organize the design to take advantage of the fastest data paths. This is most easily done by approaching the design in a sum-of-products format which then allows the design to be partitioned most effectively.

Registered Design

The considerations presented for combinatorial designs hold true for the combinatorial portion of registered designs. Registered designs can take advantage of additional features such as pipelining which will allow a design requiring several GLB levels to function as if it was implemented in a single GLB level.

Using conventional combinatorial techniques to build an Up-Counter, the pLSI family of devices will support up to 16-bits at full speed and as soon as the design requires 17 or more bits the terminal count delay will increase by 10.5 ns (t_{grp4} = 3 ns and t_{xor20} = 7.5 ns). Figure 7-27 provides a design approach which uses a registered terminal count to achieve full speed operation of a counter, any number of bits wide.

Example 14: Full Speed Wide Counters

In a conventional counter, each bit of the counter toggles when all the lower significant bits are at their terminal state. For example, in an Up-Counter, bit 5 will toggle when bits 0 through 4 are all high. When this approach is used the longest delay from any of the bits determines the maximum usable clock frequency.

The schematic presented in figure 7-25 shows a conventional 16-bit counter which shows the Product Term used at each bit. When the counter size goes beyond 16 bits the design must cascade Product Terms to generate the toggle signal for the high order bits as shown in figure 7-26. As mentioned above this adds 10.5 ns to the delay path slowing the maximum clock rate.

Figure 7-25. Conventional Counter Schematic

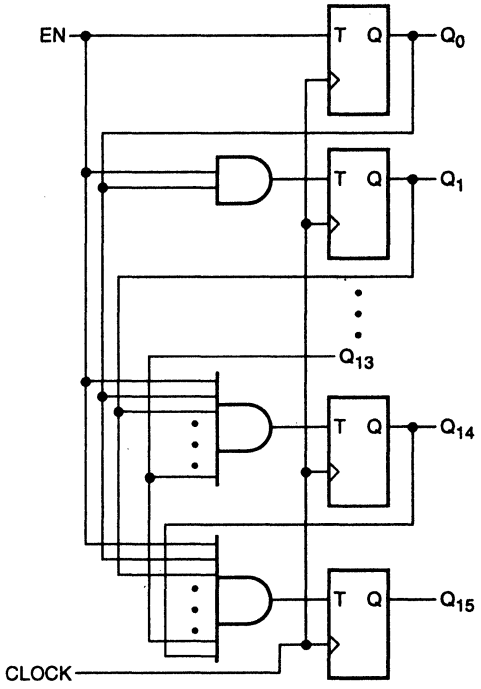
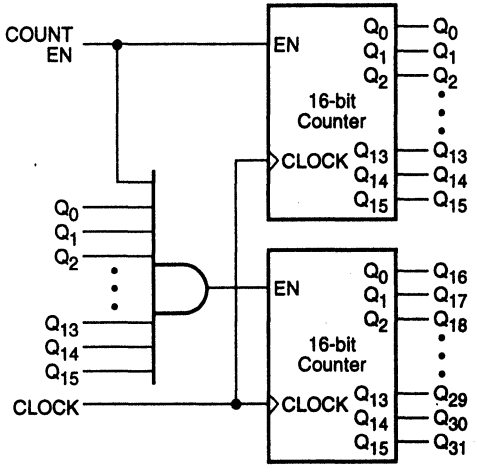


Figure 7-26. Cascading Counters



In the pipelined counter the same rules apply for toggling the next bit however, the design will be modified to pre-decode the least significant bits and then register this signal, so that it arrives at the most significant bits at the proper time.

The basic 16-bit counter will be modified slightly to provide two high speed enable signal inputs, as shown in figure 7-27. This modification provides a full speed Count Enable input as well as a full speed Carry Input.

Using this 16-bit counter the design can now be cascaded using the pipelined approach to achieve very wide counters. This is shown in figure 7-28.

The pipelined approach can be used in any regular clocked design, such as counters and many state machines.

The heart of this approach and any approach that attempts to get the maximum speed from a design is to minimize the number of GLBs the signal must go through and to use the fastest paths within the device whenever possible.

Figure 7-27. Modified 16-bit Counter

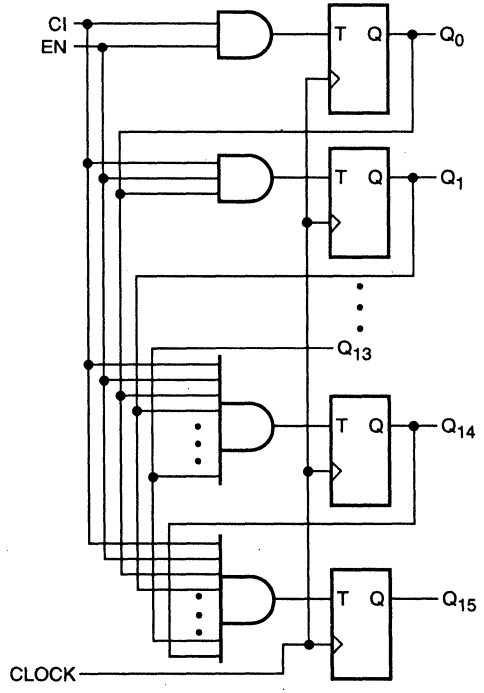
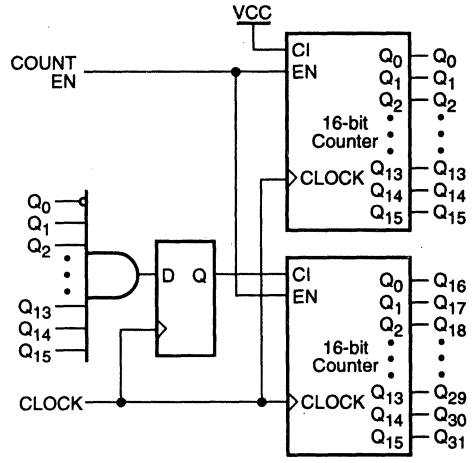


Figure 7-28. Full speed 32-bit Counter



Section 1: Introduction to pLSI™ and ispLSI™

Section 2: pLSI and ispLSI Data Sheets

Section 3: GAL® Data Specifications

Section 4: pLSI and ispLSI Architecture

Section 5: pLSI and ispLSI Advantages

Section 6: pLSI and ispLSI Software Development Tools

Section 7: pLSI and ispLSI Design Optimization

Section 8: pLSI and ispLSI Application Notes

Selecting the Right High Density Device 8-1

Beginner's Guide 8-7

A Digital Clock Design Example 8-21

ispLSI Configurable Memory Controller 8-33

pLSI and ispLSI: A Multiple Function Solution 8-47

Video Graphics Controller 8-67

Compiling Multiple PLDs into a pLSI Device 8-91

Section 9: pLSI and ispLSI Article Reprints

Section 10: pLSI and ispLSI Quality, Reliability & Programmer Support

Section 11: Sales Information

Lattice

Selecting the Right High Density Device

January 1992 **Technical Note 2004**

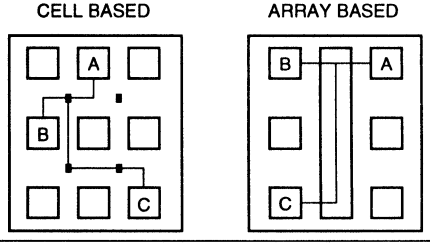
Introduction

Board designers today have several options for implementing their designs in high density programmable devices. Due to technology and design considerations, no one device is the best solution for the challenges facing designers. To address this, design engineers often use multiple types of high density devices on a single board. This paper will outline various applications issues and examine the appropriate high density solutions. It will also examine from the perspective of the user, the impact of design implementation, on the process of selecting a device.

High density programmable devices can be broadly classified into two major types; Field Programmable Gate Arrays (FGPA) and High Density Programmable Logic Devices (HDPLD). Field Programmable Gate Array devices are cell based and usually have small grain-size logic blocks with distributed interconnects across the device. High Density Programmable Logic Devices are array based and have large grained AND-OR array logic blocks with centralized interconnects (see figure 8-1) . Similarly, board designs can be broadly classified into two types: control intensive and data intensive. Control intensive designs, usually contain such subfunctions as Cache control, DRAM control, DMA control and require limited data manipulation. Data intensive designs, on the other hand, require complex manipulation of data bits which are typically found in telecom type applications. To select a high density device a designer must examine:

- ❑ Performance
- ❑ Utilization
- ❑ Ease of Use

Figure 8-1. Cell based and Array based Devices



Performance

When implementing a logic design into a high density device, it is typically partitioned into multiple logic blocks or cells and then the various cells are connected together using interconnect resources. The performance of a design is determined by the combination of the cell speed and the interconnect speed.

A logic function is divided into subfunctions which fit the basic building block of the high density device. Often the number of inputs is the most important consideration. The subfunctions should require no more inputs than are available in the logic block of the device. Smaller logic blocks tend to be faster but they offer fewer inputs. Functional implementation often requires a number of logic blocks cascaded into multiple levels of delay to implement the logic. This slows down the function speed dramatically. Control functions are typically input intensive and will be faster in devices with building blocks that allow for a large number of inputs. Data functions require fewer inputs and may be faster in devices which have fewer inputs per logic block.

Cell based devices are very granular and have very small logic blocks. They have four to eight inputs per logic block with cell speeds of 6 to 7 ns. While these devices can implement critical data functions at fast speeds, for most control functions they require 2 to 3 levels of cascading delays.

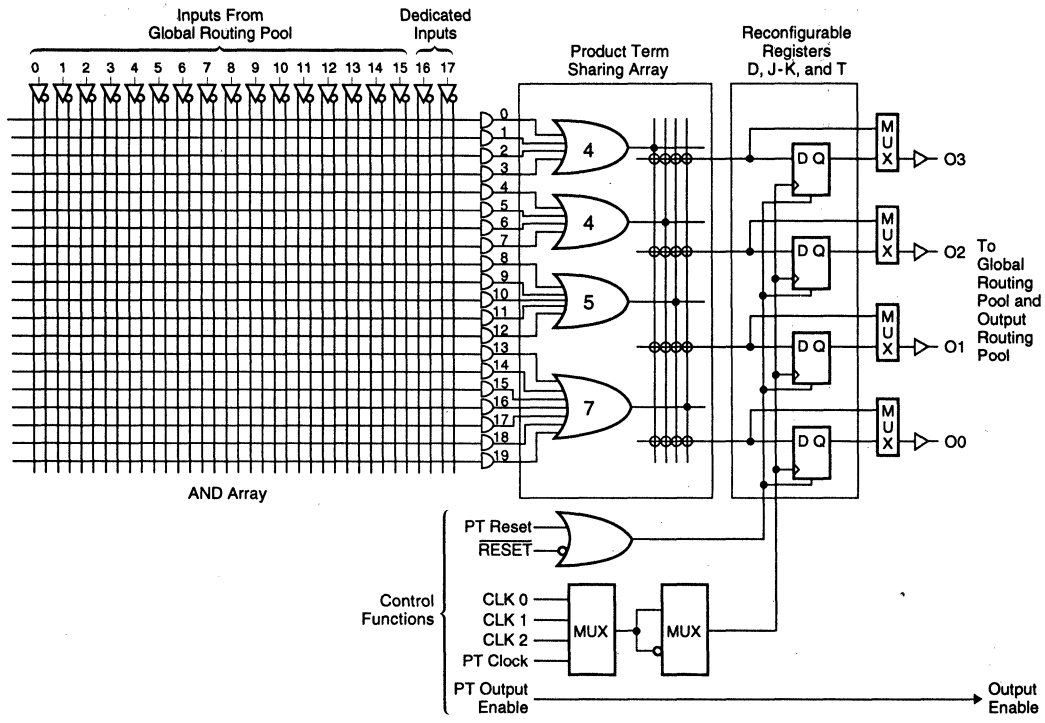
The Array based devices have larger building blocks with 16 to 48 inputs and delays of 8 to 10 ns. They can accommodate most control logic function requirements in terms of inputs to the logic block and implement them in one level of delay. However, an overly large input logic block is ineffective as it only adds to the logic block delay.

Another alternative is the programmable Large Scale Integration (pLSI™) devices (see figure 8-2) and in-system programmable Large Scale Integration (ispLSI™) devices which offer a large number of inputs (18) in every Generic Logic Block (GLB). These inputs are sufficient to accommodate the logic requirements of control functions from 8 to 12 inputs with the fastest possible speed. They also accommodate data functions which require 2 to 4 inputs per output, while maintaining high speed.

8

LATTICE SEMICONDUCTOR CORP., 5555 Northeast Moore Ct., Hillsboro, Oregon 97124, U.S.A.
Tel. 1-800-LATTICE (528-8423); FAX (503) 681-3037

Figure 8-2. pLSI GLB with 18 inputs

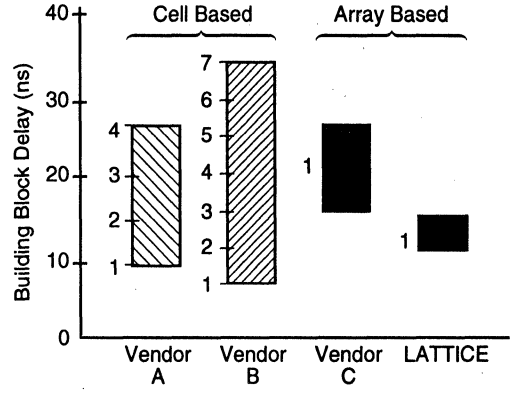


The graph illustrates (see figure 8-3) multiple logic block delays required to perform common logic functions in some of the popular high density devices available today. Due to a limited number of inputs available in Cell based devices the number of logic blocks cascaded to perform a function can be as high as 4 to 7. The Array based devices (vendor C) require only one level of delay. pLSI and ispLSI devices require only one logic block for most functions. The logic block delay is small and is comparable to most Cell based devices.

Interconnect speed is an important consideration not only for connecting subfunction logic blocks, but also for connecting signals from one logic function to another. Interconnects effect final system performance as much as logic blocks do.

The Cell based devices offer distributed interconnects with variable length lines spanning the length and the width of the device and interconnecting various logic blocks with finite delay interconnect points. Frequently, signals have to

Figure 8-3. Building Block Performance

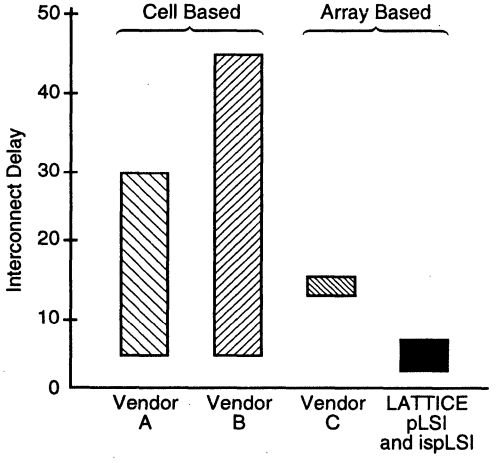


SOURCE: Lattice Applications, Vendor Literature

traverse multiple line segments and interconnect points for a connection. In general, closely located logic blocks have a shorter delay since signals travel through fewer lines and interconnect points. The opposite is true for logic blocks located further apart. There is a large variation in the interconnect delays based on the placement of the related logic blocks (see figure 8-4). In general, to improve system performance for control oriented functions in a Cell based device, a large number of signals and related logic blocks need to be placed in close proximity. Frequently, this is physically impossible and/or requires many placement iterations. Often, for such designs as state machines, counters, etc., the final performance is determined by the worst case signal speed. In such cases, the Cell based devices with distributed interconnects offer slower interconnect performance and consequently slower overall system performance. Data functions require fewer delays and can be implemented relatively faster. Placement of related data bits close to each other facilitates fast performance for data oriented functions. However sometimes with a large number of data bits this becomes difficult to achieve.

The Array based devices with centralized interconnects offer uniform interconnect delays. pLSI and ispLSI devices offer uniform interconnect delays with significantly faster interconnect performance than existing Array based devices, and consistently provide best case Cell based device delays as illustrated in Figure 8-4.

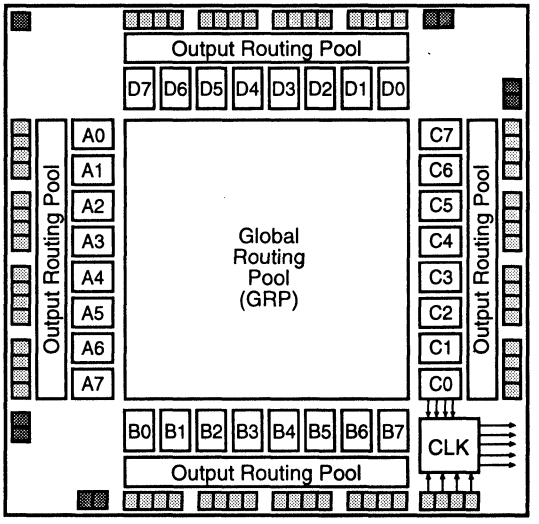
Figure 8-4. Interconnect Performance



SOURCE: Lattice Applications, Vendor Literature

The state-of-the-art for Automatic Place and Route (APR) software has not reached a point where the interconnect performance can be called optimal. In general, the Cell based devices either require a long APR time, typically a number of hours or days, in order to reach near-optimal interconnect speeds or otherwise offer mediocre performance. Uniform delays in Array based devices eliminate the need for intelligently placing related logic blocks closer, thereby reducing APR time to a few seconds. pLSI and ispLSI devices go a step further and offer faster interconnect delays using the proprietary centralized Global Routing Pool (GRP) (see figure 8-5) which retains fast APR times. This is especially good for data intensive designs where all data bits perform equally.

Figure 8-5. pLSI 1032 Block Diagram



Utilization

When mapping a design, the utilization is defined by how much of a device is used. In general, granular architectures are more effective in offering higher utilization for data intensive designs than large logic block architectures which are better for control intensive designs.

Typically Array based architectures require sophisticated synthesis algorithms to compress logic from multiple stages into single stage large blocks to increase utilization. While Array based devices implement control intensive designs more effectively, the Cell based architectures with their smaller logic blocks have less need to compress logic into one stage of the design. Data intensive designs are more

8

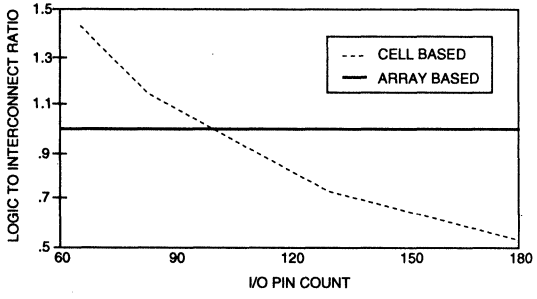
effectively implemented in a Cell based architecture which have higher register to logic ratios, since data intensive designs typically require a large number of registers. pLSI and ispLSI devices are neither as granular as some Cell based architectures, nor as large as some Array based architectures, since they offer a grain size of four outputs per logic block (GLB). However, they offer effective utilization when real life designs are considered, which are a combination of data and control functions.

Within each GLB, pLSI and ispLSI devices offer a Product Term Sharing Array (PTSA). The PTSA optionally shares GLB product terms between the four GLB outputs, enhancing logic block utilization.

As devices are scaled to higher densities, the interconnect resources should increase at the same pace as the logic resources. This ensures that all of the available logic is fully utilized in the device. The distributed nature of Cell based interconnects does not lend itself well to this scaling. Figure 8-6 shows the logic-to-interconnect ratio for one of the families of Cell based high density devices. At higher densities, this means a lower utilization of the device since the logic cannot be mapped as easily as at the lower end of the spectrum. The Array based devices scale the interconnect resources at the same level as the logic resources and offer better utilization at the higher end range of devices as well.

The pLSI and ispLSI Global Routing Pool (GRP) provides all signals globally to all device GLBs. The GRP size is scaled to provide full 1:1 logic to interconnect ratio ensuring all device logic is fully utilized, irrespective of the device size.

Figure 8-6. Logic to Interconnect Ratio



Ease of Use

Most designers use high density devices as a means for logic implementation and concentrate on the main functionality of the board, e.g., the microprocessor or the graphics section. Their time is spent on the overall functionality of the board and not on the basic logic. Ease of use and quick design turnaround times are critical to any digital designer. Ease of use is determined by a number of factors. Some critical factors directly related to the choice of device architecture are:

- Predictability of Performance
- Design Rework
- Design Entry
- Turnaround time

Predictability of Performance

The performance of the design is determined by the system considerations and is usually driven by the processor requirements or other considerations like graphics screen resolution, etc. High density devices frequently do not determine the final system speed. Designers will need to know in advance the final performance of the logic implemented in the high density device to determine the feasibility of the part selected. A designer also needs to know the speed grade required in advance, in order to estimate the cost of the design.

For Cell based devices, the number of delay levels it would take to implement the design function is not typically known. Also modifications to the design often may cause a change in the number of delay levels. Similarly, it is difficult to predict how the software will place the logic blocks as explained earlier. Even if only one out of ten critical signals ends up being slow, it will slow down the device system speed. Array based devices as well as pLSI and ispLSI devices, have predictable levels of logic and interconnect delays, which allows the designer to not only estimate speed in advance but maintain fast speeds.

Design Rework

Very few designs work the first time after entering the logic into a device. Most designs not only require logic addition or subtraction but frequently require pinout changes, and rework. This rework is often due to logic debugging and can sometimes be due to changes in the specification of the final product etc. Also a large category of digital designers

prefer an incremental design approach where small chunks of designs are implemented at a time and debugged before new chunks are added.

For Cell based devices every logic change requires a new set of logic mappings into the device cells and a new set of interconnect mapping into device interconnect lines. This leads to significant changes in the performance of the device and to undesired pinout change.

Array based devices typically do not have any adverse performance changes due to logic changes. However, pinout changes frequently occur.

pLSI and ispLSI devices were developed to allow users to make logic changes without any performance impact and to freeze pinouts when incremental design changes are done. pLSI and ispLSI devices offer an Output Routing Pool (ORP) which allows GLB outputs to be routed to many different I/O pins. Also the pLSI and ispLSI GRP allows I/O pin inputs to be available to all GLBs. These two features when combined offer the flexibility necessary to maintain pinouts in subsequent iterations of designs while maintaining the same performance.

Design Entry

There are two categories of digital designers using high density devices. The first is the designer who is a PLD user, such as with GAL devices and is familiar with Boolean, State Machine or HDL type of design entry syntax. For these designers, Array based devices offer direct mapping correlation from the entry syntax to design implementation, which is very helpful in control intensive designs. This makes such factors as the logic implementation, speed of functions and race conditions etc., predictable to the designer and simplifies the design task. The other category is the Gate Array designer who migrates to programmable Gate Array devices. For these designers the Cell based devices offer a closer correlation between schematic entry to actual design implementation. These designers also implement data intensive designs effectively, since they have a number of TTL type data function Macros available to them. With synthesis techniques however, the schematic entry is also offered for Array based devices. Familiar design entry methodology also speeds design entry time and simplifies the design process.

pLSI and ispLSI devices offer direct correlation with Boolean/HDL/State Machine entry syntax. Extensive synthesis techniques are also used in the pLSI/ispLSI Development System software to offer easy schematic capture along with a large library of the TTL Macros.

Turnaround Time

Once a design is entered, the next critical step is design compilation and programming of the part. For cell based devices with smaller logic blocks, the distributed nature of interconnects complicates matters. The Place and Route algorithm needs to satisfy multiple and often conflicting requirements for:

- Placing Related Logic Closer for Faster Speed.
- Moving Logic to Satisfy Critical Timing Requirements.
- Moving Logic Due to the Lack of Interconnect Resources.
- Repartitioning Logic to Satisfy the Above Three Conditions.

These four basic requirements are interrelated and complex, making the compilation process very time consuming. Typical Cell based devices require 2 to 8 hours for compilation in order to achieve reasonable system performance objectives.

The Array based devices with global connectivity and uniform interconnect delays eliminate the need to closely place related logic. pLSI and ispLSI devices with centralized interconnect offer compilation times of minutes versus hours for the Cell based devices, improving designer productivity. This combined with the ispLSI version of the family which allows on-board reprogramming of multiple devices simultaneously, offers a whole new dimension for logic design.

Conclusion

A digital designer has multiple choices available for high density designs. The current solutions broadly categorized as Cell based and Array based devices, offer alternative advantages and disadvantages for digital designers. The type of solution chosen depends upon the type of functions being implemented, the performance required and other design specific trade-offs. pLSI and ispLSI devices offer the advantages of both Cell based and Array based devices.

High Performance - pLSI and ispLSI devices are designed to be extremely fast for both control and data intensive functions and are particularly excellent for functions requiring more than eight inputs per logic block.

Predictable Delays - The centralized GRP structure combined with wide input GLBs offers uniform delays which allows the designer to determine system speed in advance as well as maintain constant speeds in subsequent iterations of the design.

High Utilization - The ability to scale interconnect resources at the same level as logic resources combined with built-in flexibility of the GRP and ORP assure high device utilization. Also the PTSA adds another level of flexibility for increasing logic block utilization.

Ease of Use - Predictable performance, quick design entry and rework time provide fast design turnaround. This simplifies the design process and enhances time to market.

The combination of high performance, predictable delays, high utilization and ease of use, not only offers a superior solution for design requirements; it is delivered in E^2CMOS technology with reprogrammability and 100% testability offering unparalleled device quality.

Lattice Beginner's Guide

January 1992 **Application Note 2500**

Introduction

This Beginner's guide is designed to help you become familiar with the Lattice pLSI™ 1032 device, ispLSI™ 1032 device and the Lattice pLSI/ispLSI Development System (pDS™). To do this, a complete design of a simple four-bit counter is discussed from specification through programming and testing the part. The following assumptions are being made. First, you have read and understood the pLSI 1032 data sheet. Next, you have the documentation for Microsoft® Windows™ readily available. Everything else should be here in this Beginner's Guide.

The Lattice pDS software is designed to run under Microsoft Windows Version 3.0 (see figure 8-7).

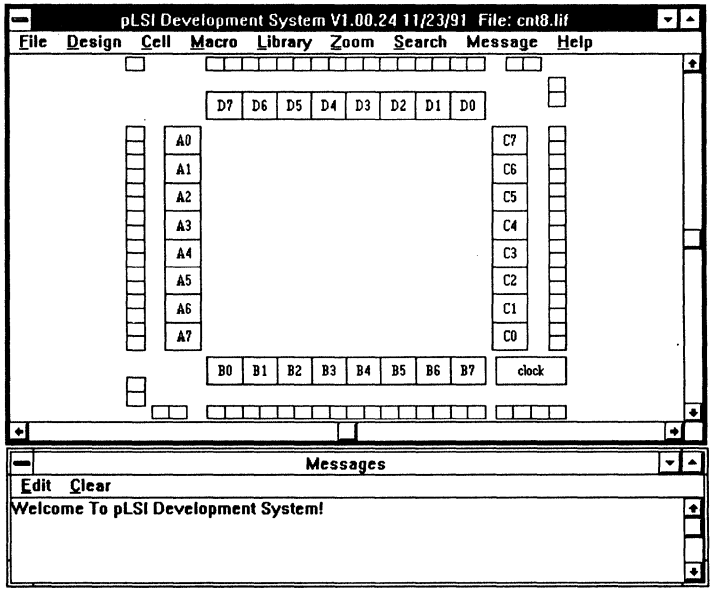
Windows is an industry standard Graphical User Interface (GUI) for pull down menus, text editing commands and screen control commands. Because the Lattice interface is the same as other Windows programs, it is very easy to learn. If you know how to run any Windows program, you can run the Lattice software.

It is necessary to have Windows for the Lattice pDS software to run. Windows runs on most standard IBM PCs or clones. If your computer runs Windows 3.0, it will run the Lattice pDS software. The recommended system configuration for running pDS software is:

- ❑ A 386 or 486 Processor
- ❑ 4 Megabytes of RAM
- ❑ 40 Megabyte Hard Disk
- ❑ A Floppy Disk Drive
- ❑ A Microsoft Windows Compatible Mouse
- ❑ VGA or Super VGA Graphics

In addition, the pDS software requires that either a spare parallel printer port be available to perform in-system programming, or a spare serial port be available to communicate with an RS-232 controlled programmer.

Figure 8-7. Lattice pDS software Opening Screen



LATTICE SEMICONDUCTOR CORP., 5555 Northeast Moore Ct., Hillsboro, Oregon 97124, U.S.A.
Tel. 1-800-LATTICE (528-8423); FAX (503) 681-3037

Getting Started

If you have not previously installed the Lattice pDS software, see the installation procedure which came with your development system.

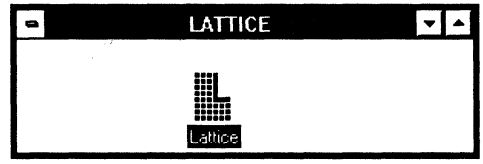
1. To start Windows, type WIN at the DOS Prompt (C:>).
2. Install the Lattice pDS software according to the installation instructions. A new program group called LATTICE is created. This program group should contain a single icon, called LATTICE.EXE, which looks like the Lattice company logo (see figure 8-8).
3. To start the pDS software, double click on the Lattice Logo Icon.

Before you can proceed any further, some of the Microsoft Windows tasks that you should be able to perform are:

- ❑ Selecting a Menu Item Using the Mouse
- ❑ Using *Open*, *Save* and *Save As* Menu Items
- ❑ Entering Commands and Text into Message Windows and Dialog Boxes
- ❑ Moving Around the Screen with the Scroll Bars
- ❑ Editing Text Using the Keyboard and Mouse to:
  - Select the Insertion Point
  - Select Text by Highlighting It
  - Cut, Paste and Copy Text

If you are unfamiliar with any of these options, then take some time to go through the Windows Users Guide. If you have ever worked with the Apple™ Macintosh™, you will find that many of the commands and operations are similar.

Figure 8-8. Lattice Program Group Window



A Brief Tour of the Screen

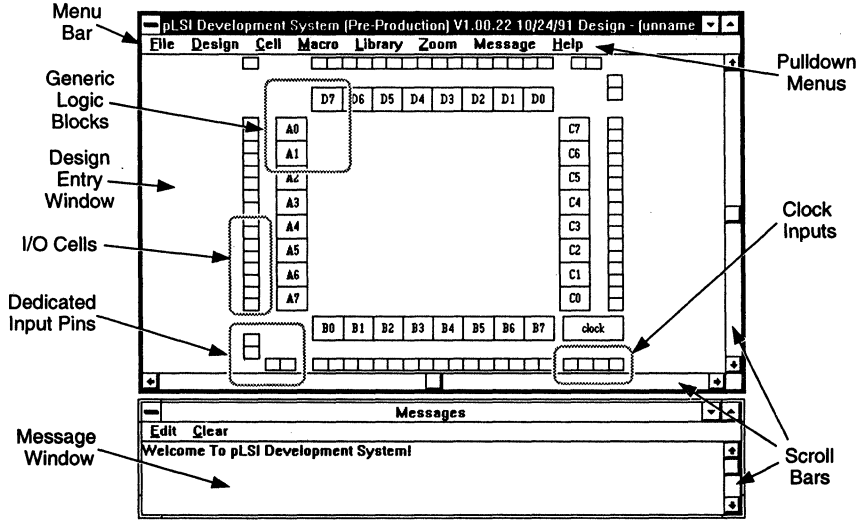
Once you invoke the Lattice pDS software, two windows are displayed (see figure 8-9).

The larger of the two windows displays a graphical representation of the pLSI 1032 logic diagram. This window is called the Data Entry Window. The design is entered by editing equations in the Data Entry Window.

The smaller of the two windows is the Message Window and it is located at the bottom of the screen. The pDS software communicates with you by placing messages in the message window.

The part that is displayed in the block diagram shows the elements of the pLSI 1032 that can be modified by the user. These elements are the GLBs, the I/O Cells, the dedicated input pins, and the clock input pins, as indicated in Figure 8-9.

Figure 8-9. Design Entry Window



The design is entered into the development software by clicking on the block that you wish to edit and entering equations or Macros into the Edit Window that appears (see figure 8-10).

The method of entering the configuration data into a cell depends on what type of cell it is:

Configuration data for Generic Logic Blocks is entered using a combination of Boolean Equations or Macros from the Lattice Standard Library.

Configuration data for I/O Cells is entered using Macros only. There is a complete set of Macros which describes all possible combinations of input, output, and I/O cell configurations.

Configuration data for the Dedicated Input Pins and the Clock Input Pins is entered using a subset of the I/O Cell Macros. Because these pins are inputs only, and do not have input registers, many of the standard I/O Cell Macros cannot be used.

The Design Flow

Before starting our sample design, it is valuable to understand the Design Flow. The following steps are observed to complete a design. Refer to figure 8-11 for more information.

Specifying the Design

A design is specified using one of the two approaches. With the first method, you use an existing design, consisting of 7400 Series TTL Logic elements, and fit the design into the pLSI part. With the second method, you design a circuit that is optimized for best performance and utilization of the pLSI architecture.

Figure 8-10. Open Edit Windows

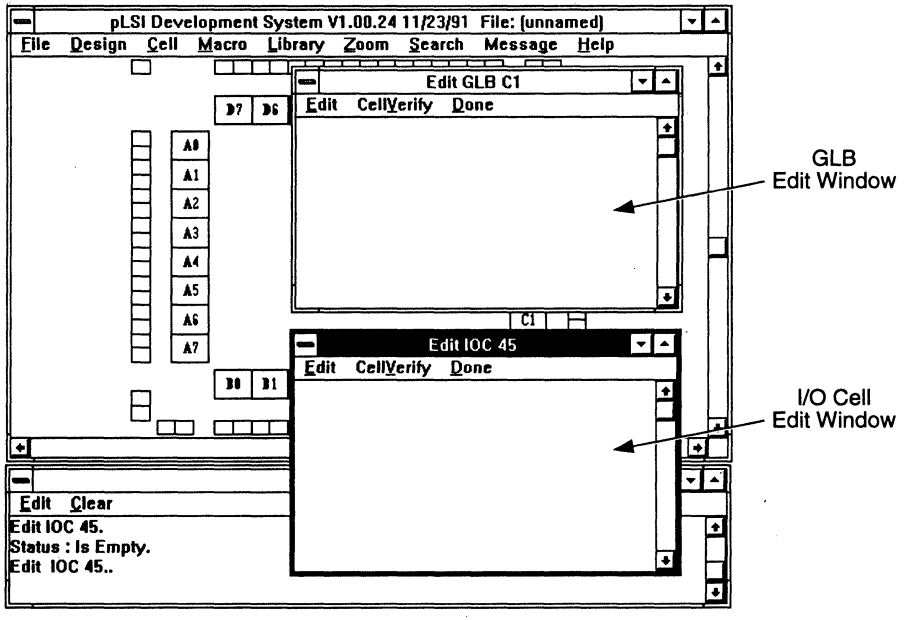
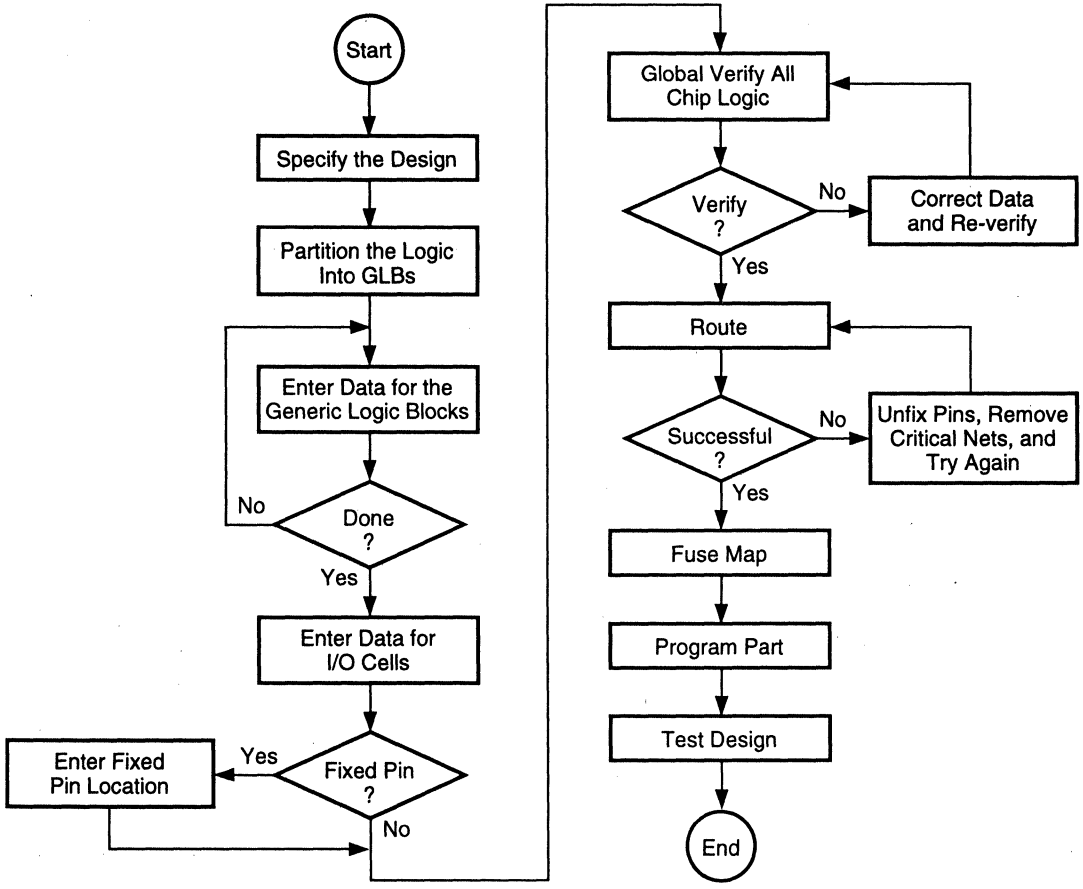


Figure 8-11. Design Process Flow



The first approach consists of simply selecting Macros from the Lattice library that approximate the functions of the TTL or CMOS circuits and then connecting them to each other. Using this approach a design can be completed quickly and has a high degree of probability of working the first time because the circuit has been tested.

The second approach ensures better performance and higher utilization, but may require some circuit redesign. Many designs are a combination of the two approaches.

To select the correct pLSI device, partition the design into GLBs, and count the number of GLBs and I/O Cells used. Next, select the pLSI device that can hold the amount of logic required. Selection of the proper device is based on the amount of logic required and on the number of I/O cells needed.

The best utilization and routability are achieved by allowing the software to assign the I/O pin placement. It is a good idea to design the pLSI part first, and then lay out the printed circuit board or wire-wrap board after the device has been routed. Once the software intelligently assigns the pin placement the first time, the pins can be fixed, and changes can be made to the logic with few problems.

Partitioning the Design

Partitioning consists of carving the logic into chunks that conveniently fit into the pLSI Generic Logic Blocks. These general rules should be followed when partitioning logic:

- Look at the Macro library and decide if any of the logic can be implemented using the standard Macros. Macros are already partitioned and are optimized for high utilization and high performance. Macros are also the fastest method to input the logic design.

- Know the capabilities of the GLB. It has 18 Inputs and 4 Outputs. The GLB has 20 Product Terms that are grouped together in groups of 4, 4, 5, and 7 PTs. The registers in the GLB share a common clock. The registers within the GLB also share a common Reset Product Term.
- When an output has been fixed to a specific I/O pin, the signal that is used to generate that output must be generated within the same Megablock.
- There is only one Output Enable signal per Megablock. Outputs which share a common Output Enable signal should be placed in the same Megablock.
- Signals that are related to each other, such as counters, shift registers, etc., should be placed into the same Megablock. This is done to reduce routing congestion.

Compiling the Design

Compiling the design is done using the Lattice pDS software and consists of four steps:

1. **Entering the design.** Boolean equations or Macros are entered into the various cells and blocks on the pLSI device using a built in text editor. After each cell has been entered, a *Local Verify* is done to check for syntactical or logical errors within that cell.
2. **Verifying the design.** This is done globally after all the design has been entered. This verification looks for such problems as inputs that are not connected to the GLBs or nets that have duplicate names. The design must completely pass a Global Verify before any of the following steps can happen.
3. **Routing the design.** This is the next step after a successful Verify. The Router interconnects the Generic Logic Block and I/O Cell inputs and outputs. The option of fixing certain input and output signals to specific device pins is available.
4. **Generating the Fusemap.** This takes the verified and routed design and creates the *JEDEC* fuse file necessary to program the part. This is a modified format JEDEC file, and the file generated has a suffix of .JED.

Programming the Part

Once the design has been compiled, the next step is to program the part. This can either be done on the board if using in-system programming (isp) or in a separate programmer. Using a separate programmer requires that the part be removed from the target system socket and inserted into the programmer to program the part.

Testing the Design

The last step in the process is testing the design. The design is tested by putting it on the board and seeing if it works correctly. If corrections need to be made, the appropriate GLBs or I/O Cells are reprogrammed, and the design is recompiled. Because the pLSI 1032 is an electrically erasable and reprogrammable part, the same part can be used again.

The Sample Design

The sample design is a simple one. We are going to design a 4-bit binary counter using Boolean equations and place it into a pLSI 1032 device. We will then take the design through the compilation process, generate a fuse file and program a part.

The counter has the following specifications:

- A 4-bit Synchronous Binary Counter.
- An Active High Cascade In (CI) and Cascade Out (CO) Pins.
- An Active High Count Enable (CE) Pin.
- A Synchronous Reset Pin.

Figure 8-12 shows the schematic diagram and Figure 8-13 shows the logic symbol for this counter. Because the counter has 5 outputs (Q0, Q1, Q2, Q3, and Cascade Out) it occupies two GLBs.

In this design example, the Clock and I/O pins are assigned to be compatible with the Lattice ispLSI 1032 Demonstration board. This allows the design to be tested easily.

The input signals Cascade In, Count Enable and Reset are connected to three bits of the 8-bit DIP switch, and the five outputs are connected to five of the discrete LED outputs.

Defining the Counter

In defining the counter, the first step is to write the equations. The equations for the 4-bit binary counter are expressed in Listing 1.

There are two inputs to the Exclusive-OR gate in front of the D input to the register. We shall call the one that receives its input from the feedback of the same register as the data input. It is to the left of the $$ (XOR) symbol in the above equations. The other input is connected to the control terms Cascade In and Count Enable. These are called the control input. When the control input to the XOR is a zero the output of the XOR follows the data input (Hold.) When the control Input is a one, the output of the XOR is inverted from the input (Increment.)

Figure 8-12. Counter Schematic Diagram

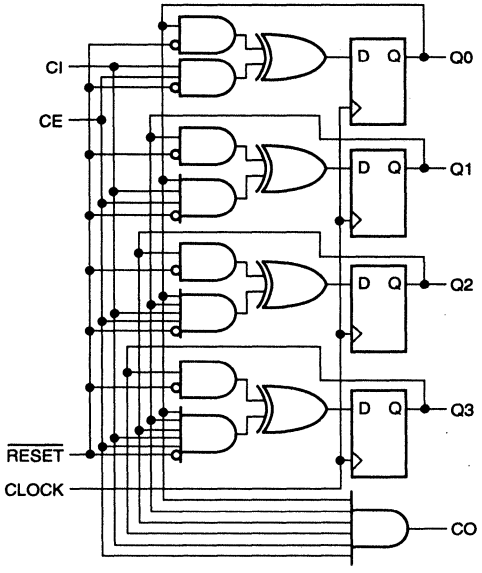
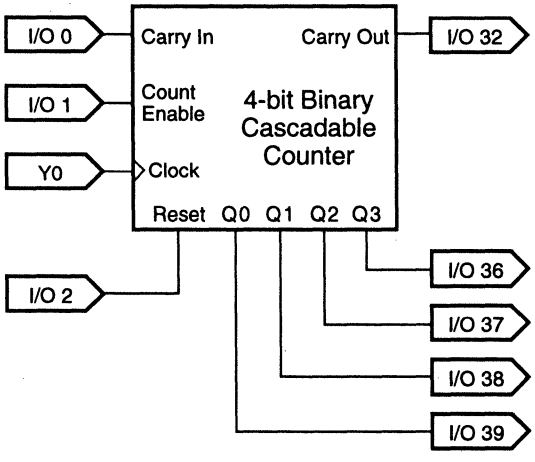


Figure 8-13. Sample Cascadable Counter Logic



When either Cascade In or Count Enable is low and RST is low, the Q0 output from the counter remains in its current state (Hold). When Cascade In and Count Enable are both high and RST is low, the Q0 output toggles on each successive clock (increment.) When RST goes High, the inputs to the Data side of the XOR gate and the Control side go low. This causes the output of the counter to go low on the next clock edge (Reset.)

Each successive stage operates similarly, except during transition, (Increment), when the outputs of all previous stages are at logic level one. The Carry Out signal is only generated when all the stages have reached a one and both Cascade In and Counter Enable are a one.

Once the equations have been defined, enter them into the GLBs. Follow these steps:

1. From within Windows, start the Lattice pDS software by double clicking on the Lattice Icon.
2. When the Lattice software starts, it displays the block diagram of the pLSI 1032 part. Open GLB C1 for editing by double clicking on it. The edit window displays.
3. Enter the equations shown in Listing 2 into the edit window for GLB C1.
4. Verify the equations by clicking on the Cell Verify menu option. If errors appear in the Message window, find out what is wrong, and correct it. Things to look for are typing errors, missing semicolons, or incorrect symbols. Re-verify after making corrections.
5. Close the Edit window for GLB C1 by selecting the Done option from the Cell Edit Menu.
6. Open GLB C2 for editing by double clicking on it.
7. Enter the following equations into the edit window for GLB C2:

```
SIGTYPE  CO  OUT;

EQUATIONS
    CO = Q0 & Q1 & Q2 & Q3 & CI & CE;
END
```

8. Verify the equations by clicking on the Cell Verify menu option.
9. Close the Cell Edit window by clicking on the Done option in the menu bar. See Figure 8-14.

At this point, the logic for the counter is completely specified, but we still must connect the Clock and the Inputs and Outputs.

10. Open Clock Input Y0 by double clicking on it. It may be necessary to Zoom in on the Clock area of the Logic Diagram to determine which pin is Y0.
11. Enter the following equations into the edit window for Clock Input Y0:

```
XPIN  CLK  X_CLK  LOCK  20;
IB1  (_CLK,  X_CLK  );
```

12. Verify the equations by clicking on the Cell Verify menu option.
13. Once the cell verifies correctly, close the Cell Edit window by clicking on the Done option in the menu bar.

Listing 1. Counter Equations

```
Q0 = (Q0 & !_RST) $$ (CI & CE & !_RST)
Q1 = (Q1 & !_RST) $$ (Q0 & CI & CE & !_RST)
Q2 = (Q2 & !_RST) $$ (Q0 & Q1 & CI & CE & !_RST)
Q3 = (Q3 & !_RST) $$ (Q0 & Q1 & Q2 & CI & CE & !_RST)
CO = Q0 & Q1 & Q2 & Q3 & CI & CE
```

Listing 2. GLB Equations

```
SIGTYPE  Q0  REG  OUT;
SIGTYPE  Q1  REG  OUT;
SIGTYPE  Q2  REG  OUT;
SIGTYPE  Q3  REG  OUT;
EQUATIONS
    Q0.CLK = _CLK;
    Q0 = (Q0 & !_RST) $$ (CI & CE & !_RST);
    Q1 = (Q1 & !_RST) $$ (Q0 & CI & CE & !_RST);
    Q2 = (Q2 & !_RST) $$ (Q0 & Q1 & CI & CE & !_RST);
    Q3 = (Q3 & !_RST) $$ (Q0 & Q1 & Q2 & CI &  CE & !_RST);
END
```

14. Repeat Steps 10 through 13 for the Cascade In input pin located at I/O 0 using these equations:

```
XPIN IO XCI LOCK 26;
IB1 (CI, XCI);
```

15. Repeat Steps 10 through 13 for the Count Enable input pin located at I/O 1 using these equations:

```
XPIN IO XCE LOCK 27;
IB1 (CE, XCE);
```

16. Repeat Steps 10 through 13 for the Reset input pin located at I/O 2 using these equations:

```
XPIN IO X_RST LOCK 28;
IB1 (_RST, X_RST);
```

17. Repeat Steps 10 through 13 for the Q0 output pin located at I/O 39 using these equations:

```
XPIN IO XQ0 LOCK 75;
OB1 (XQ0, Q0);
```

18. Repeat Steps 10 through 13 for the Q1 output pin located at I/O 38 using these equations:

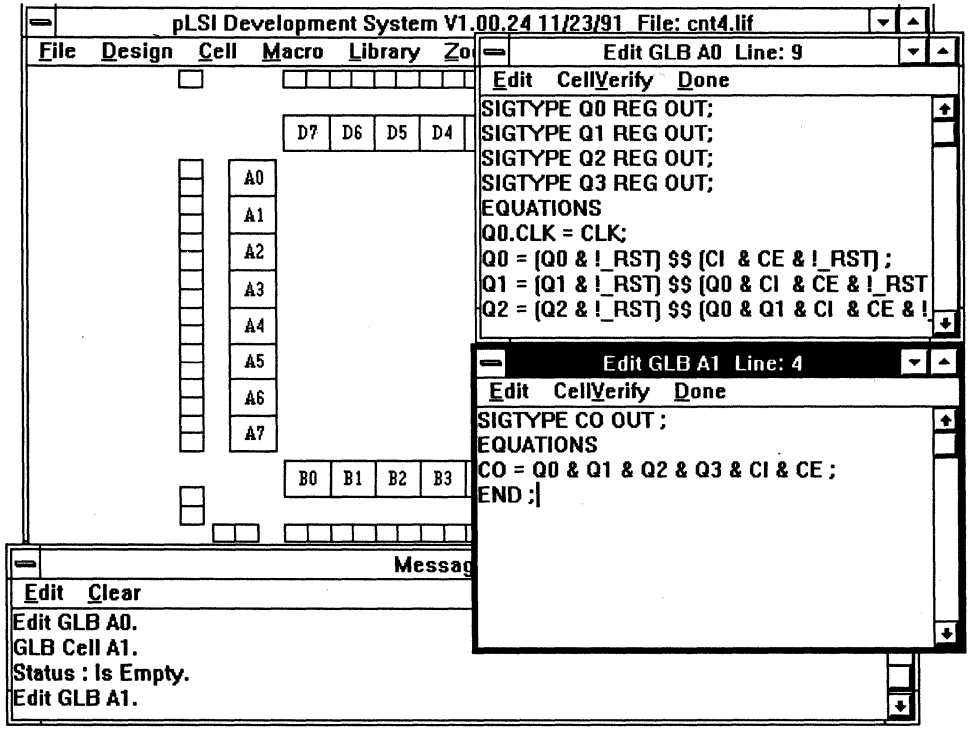
```
XPIN IO XQ1 LOCK 74;
OB1 (XQ1, Q1);
```

Note: With the Lattice pDS software you can have two separate Edit Windows open at the same time. This means that you can Copy the equations from I/O Cell 39 and Paste them into I/O Cell 38. The data in both cells is similar, and you can use the Windows editing commands to make changes.

19. Repeat Steps 10 through 13 for the Q2 output pin located at I/O 37 using these equations:

```
XPIN IO XQ2 LOCK 73;
OB1 (XQ2, Q2);
```

Figure 8-14. Cell Entry Windows with Counter Equations.



20. Repeat Steps 10 through 13 for the Q3 output pin located at I/O 36 using these equations:

```
XPIN IO XQ3 LOCK 72;
OB1 (XQ3, Q3);
```

21. Repeat Steps 10 through 13 for the Carry Out output pin located at I/O 39 using these equations:

```
XPIN IO XCO LOCK 68;
OB1 (XCO, CO);
```

    Now, the Inputs, Outputs and Clocks are connected, and the equations for the counter have been entered and verified. The design is complete and ready to be Globally Verified. Before proceeding, save the work.

22. From the Menu Bar, select the File Option, and choose Save As. The pDS software prompts you for the name of the file that you are saving. Type in the name COUNTER. The suffix .LIF (Lattice Internal File) is automatically appended.

The next step in the development process is to Globally Verify the integrity of the design. Global Verify first performs a Cell Verify on GLBs or I/O Cells which have not already been verified, or which have changed since the last Cell Verification. Then it checks interconnections between the GLBs looking for problems such as outputs which are not used or inputs that are not connected.

23. From the Design Menu option in the Menu Bar, select Verify. This starts the Global verify process. If Verify finds any problems, it lists them in the Message window at the bottom of the screen. The verifier also creates a netlist file that the Router uses to route the design. Once the design passes verify, it is ready to be routed.

24. From the Design Menu option in the Menu Bar, select Route. This module places I/O pins that have not previously had their positions defined, and interconnects all the logic blocks and I/O cells on the device. When Route is invoked, a list of all the I/O pins displays. If you have not previously defined which signals are connected to which pins, this is the time to do it.

25. From within the Route Message Window, click on the Execute button. This starts the router. Routing is an entirely automatic process, and requires no intervention. As before, if any problems occur, they are listed in the message window.

26. The last step in the compilation process is to generate the Fusemap. This is accomplished by clicking on the Fusemap option in the design menu. Like Route, Fusemap is an entirely automatic process, and should require no intervention. The output from the Fusemap program is the .JED file and it is used to program the part.

The design is now complete. Because it was given a name previously (COUNTER.LIF) you can simply click on the Save command in the File menu to save the work. All that remains is to program the part and test the design.

Review of the Syntax

This is a brief review of the syntax used in the example design. For complete information see the Language Reference section of the Software Manual included with the Lattice pDS software.

The operators that the Lattice pDS software uses are similar to those used by the Data I/O ABEL program. The operators and an example of how they are used are shown in the table below. The Precedence of Evaluation is also shown where 1 is the highest precedence. See table showing Precedence of evaluation.

Table : Precedence of Evaluation

Operator	Precedence	Description	Example
!	1	NOT	!A
$$	2	XOR (XOR Gate in GLB)	A $$ B
&	3	AND	A & B
#	4	OR	A # B
$	5	XOR (Soft)	A $ B
!$	5	XNOR (Soft)	A !$ B

Review of the Syntax (Continued)

In addition to the equations, there are several other lines that need to be included in the GLB or I/O Cell definition. They are:

SYM; The symbol line consists of five parts:

- The Keyword SYM that indicates what type of line this is to be.
- The Symbol Name. This is either GLB or IOC.
- The Cell location.
- The Symbol Level used by other software packages. For our purposes, always use a 1.
- The Symbol User Name. This is an assigned name that appears in the GLB or IOC in place of its location designation.

SIGTYPE; Used to define signal attributes within a GLB.

OUT defines a combinatorial output.

REG OUT defines a Registered Output.

EQUATIONS; Indicate the start of the Equation Section for a GLB or I/O Cell.

MACRO; Indicates the usage of a Macro Logic Element from the Macro Library.

END; Signifies the end of an Equation Section, a GLB or I/O Cell definition, a Declaration

Section, or a Macro Definition. There can be more than one END statement in a GLB.

Comments are indicated by preceding the comment with two forward slashes:

```
// This is an example comment line.
```

Programming the Device

The Fusemap program generated a fuse (.JED) file which needs to be permanently programmed into a pLSI 1032 or an ispLSI 1032 device. Programming the part is done using one of three methods:

- ❑ RS-232 Link Programmer for pLSI and ispLSI device
- ❑ in-system program for ispLSI device
- ❑ Motherboard Programmer for ispLSI device

For a programmer that is controlled by a serial RS-232 link, the Lattice pDS software can call up the Windows Terminal Program. By using your PC to emulate a terminal, you can give the programmer the commands necessary to set it up to receive the .JED file. The Download command in the Windows Terminal program transfers the file to the programmer. Because the .JED file is an ASCII format, a text download is used.

An isp Download Cable is offered as an option with the pDS software . The cable connects to the Parallel Port on a PC and controls the programming process. If the target system is designed to use in-system programming, the part can be programmed right on the board. The Lattice ispLSI Demonstration board is designed to allow this.

For programmers using a board that plugs into one of the slots on the PC motherboard, there is no automatic download procedure provided with the Lattice software. You have to exit Windows and start the programmer executive program.

For programming the ispLSI 1032 part, follow these commands.

1. From the Design Menu select the Program Option. This invokes the In-system programming module.
2. The isp module prompts for the name of the JEDEC file. Click on COUNTER.JED in the file list and then click on OK. It may already have COUNTER.JED as the default file name. If this is so, then just click on OK.
3. Programming takes a few seconds. If any errors are encountered, they are listed in the message box.

When programming is complete, the part is reset and sent back into the operating mode. It can then be tested by applying the required inputs and looking at the outputs.

Advanced Design Concepts

Working with Macros

The Lattice pDS software comes with a library of over 200 Macro logic elements. These logic blocks are similar to 7400 TTL logic. Some example Macros are listed in Table 8-1.

For complete information on the Macro Library refer to the Macro Reference Manual that comes with the Lattice pDS software. In addition to using Macros from the Lattice library, you can either create custom Macros from scratch, or modify Macros from the Lattice library to satisfy design requirements.

We are going to take a Macro from the library that is identical to the counter just created, and cascade it with the counter. The Macro element that is used to do this is named CBU24. The schematic diagram is shown in figure 8-15.

1. Read in the previous design using the File Read command. The name of the file is COUNTER.LIF.
2. Invoke the Macro window by clicking on MACRO in the Menu Bar.
3. Select the Macro *CBU24* from the list of Macros.
4. Click on GLB C3 to Select it.
5. Click on the *PLACE* command in the Macro Menu. This places the first half of the 4-bit counter Macro in GLB C3. The signal names that were placed in the GLB are the default signal names, and need to be changed to correspond to the signal names that used so that the router is able to connect them.
6. The original text in the cell was:

   ```
   CBU24_2(CAO,[Q0..Q3],CAI,EN);
   ```

   Change that to read:

   ```
   CBU24_2(CAO,[Q4..Q7],CO,CE);
   ```

   The default signal names are changed to match those already used in as shown in Table 8-2.
7. Perform a Cell Verify to ensure that no errors were introduced.
8. Click on *DONE* to close that GLB.
9. The Macro occupies two GLBs, so the second half of the Macro now needs to be placed. Click on *GLB C4 to place the second half.*

Table 8-1. Macro Logic Element Examples

Macro Name	7400 Part Eqivalent	Description	Number of GLBs Used
AND2	7408	2 Input AND Gate	1/4
XOR2	7486	2 Input Exclusive OR Gate	1/4
FJK21	74112	J-K Flip-Flop with Asynchronous Clear	1/4
CBU34	74161	4-Bit Prelaodable Binary Counter with Reset	1 1/4
BIN27	74247	BCD to 7 Segment Decoder	2
SRR38	74166	8-Bit Parallel In-Serial Out Shift Register	2
ADDF4	74283	4-Bit Full Adder with Look Ahead Carry	4 3/4

Figure 8-15. Custom Binary Counter Cascaded with a Standard Macro Counter

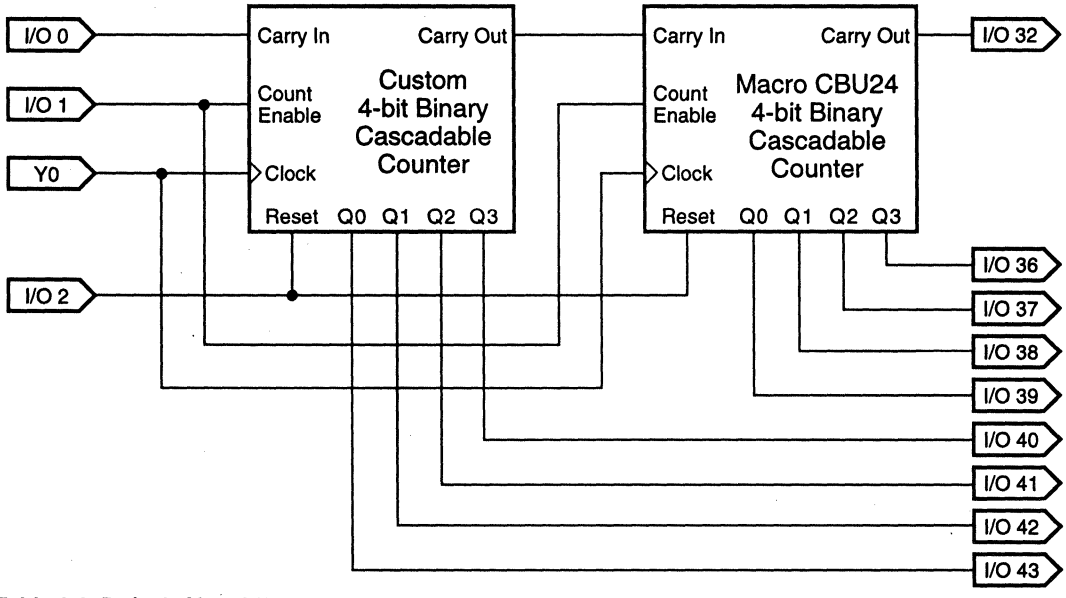


Table 8-2. Default Signal Names

Default	Signal	Is	Notes
CAO	Cascade Out	CAO	This is a new signal. We can use the default name.
Q0..Q3	Cntr Outputs	Q4..Q7	We used Q0 through Q3 in the first counter. We need to assign new names so the router won't get confused.
CAI	Cascade In	CO	CO is the name that we assigned to the Cascade Out pin on the counter that we designed.
EN	Enable	CE	We called our Enable pin CE (Count Enable). This comes from a pin external to the device.

10. As before, the signal names that were placed in the GLB were the default names. They also need to be edited. The Lattice software placed the following code into the cell:

    ```
    CBU24_1([Q0..Q3],CAI,CLK,EN,CD);
    ```

    Change it to read:

    ```
    CBU24_1([Q4..Q7],CO,_CLK,CE,RST);
    ```

    As before, we have changed the default signal names to match those that we are already using. See Table 8-3.

11. As before, perform a *CELL VERIFY*, and click on *DONE* when through.

    The counter has now been placed, and the inputs connected, but the outputs are still floating. Connect them to the I/O Cells as you did with the previous counter.

12. Select the Macro called *OB1* from the Macro list.

13. Click on IO Cell *#40* to select it.

14. Click on PLACE in the Macro window. This configures I/O Cell #40 as an output buffer, but it used the default signal names. The text that was placed in the cell was:

    ```
    OB1   (XO0,A0);
    ```

You should change it to read:

```
OB1   (XO4,Q4);
```

Q4 is the name of the first output of the counter. XO0 was changed to XO4 so that there would not be duplicate I/O cell names when we place the next cells.

15. Click on IO Cell *#41* to select it.

16. Click on *PLACE* in the Macro window. Change the default signal names to match those used in your design:

```
OB1   (XO0,A0);
```

Becomes:

```
OB1   (XO5,Q5);
```

17. Use the same technique to connect I/O cell #42 to counter output Q6.

18. Use the same technique to connect I/O cell #43 to counter output Q7.

All the outputs are now connected, and the design is complete. As in the first design, you now need to do a *Global Verify* on the design, *Route* the nets and generate the *Fusemap*. You can see from this exercise how much simpler it is to complete a design when using Macros.

The use of Macros is not limited to those in the Lattice Macro library. Sometimes the standard Macro is close to, but not exactly what you need. You can copy any of the standard Lattice soft Macros into a personal library, and modify them to meet specific needs. You can also create Macros using Boolean equations and save them in your personal library for future use.

Conclusion

We have tried to give a feeling of how to design using pDS software from definition to completion. In this Beginner's Guide, we:

- ❑ Looked at the Lattice pDS software and its various elements.
- ❑ Explained the design flow from beginning to end.
- ❑ Looked at the syntax needed for entering a design.
- ❑ Defined a small counter and partitioned it into GLBs.
- ❑ Entered the design for that counter into the development system.
- ❑ Took that design through the compilation process. (Verify, Route, and Fusemap).
- ❑ Programmed a part.
- ❑ Tested the design.
- ❑ Changed the design, and introduced the use of Macros.
- ❑ Recompiled that design and tested it.

From this you can see how simple it is to design using the Lattice pLSI and ispLSI families. If you have followed all of these steps, then you are ready to complete a design of your own.

8

Table 8-3. Renaming Default Signal Names

Default	Signal	Is	Notes:
Q0..Q3	Cntr Outputs	Q4..Q7	We used Q0 through Q3 in the first counter. We need to assign new names so the router won't get confused.
CAI	Cascade In	CO	CO is the name that we assigned to the Cascade Out pin on the counter that we designed.
CLK	Clock	_CLK	We named the signal that we brought in on pin Y0 _CLK.
EN	Enable	CE	We called our Enable pin CE (Count Enable). This comes in from pin 27.
CD	Clear Direct	RST	Our reset signal was brought in on pin 28, and called RST.

A Digital Clock Design Example

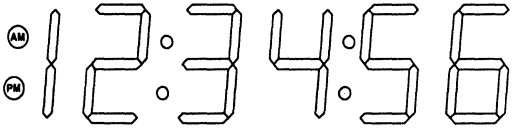
January 1992 **Application Note 2501**

Introduction

The Lattice pLSI and ispLSI families feature a fast, flexible, register intensive architecture which is suited to many applications. The pLSI and ispLSI software tools are especially useful for design because they incorporate the PLD-like Boolean syntax and eliminate many of the problems associated with Gate Array design, such as expensive design systems, extended simulation times, long lead times for first samples and long rework times.

The intent of this application note is to show how easy it is to design with an ispLSI 1032 device by implementing a simple design using many of the features of the device and design software. The digital clock was chosen because its operation is understood by most designers. This example concentrates on the design process rather than the design itself. The design also fits easily into the ispLSI 1032 demonstration box which makes it easy to debug and demonstrate. Figure 8-16 shows an example of a digital clock design.

Figure 8-16. Digital Clock Design Example



In implementing this design advanced features are used to demonstrate the flexibility of the design environment. With the pLSI Development System (pDS) Software, it is simple to do a complete design using Macro library elements which are similar to parts from a typical 7400 TTL Data Book. The logic in a Macro can also be modified to meet exact design requirements. At the other extreme for complete control over the logic within the device, the circuit may implemented with Boolean Equations. Once a custom circuit is created it can be saved as a Macro in a personal Macro library for future use.

It is assumed that the reader has read the data sheet and understands the architecture of the ispLSI device. Reading the pDS Software manuals makes it easier to understand what is being presented, but is not necessary.

The tools used in implementing this design are:

- The pDS Software Running Under Windows™ 3.0 on an IBM™ Compatible PC
- The in-system programming Cable (Download Cable) optional with the pDS Software

Entering & Compiling the Design

Before discussing details of the clock design, the following is a quick review of the design flow. In the pDS software, designs are created using either Boolean Equations or Macros taken from the Lattice Macro library.

Boolean logic is utilized because it is easy to use. The syntax used is similar to that used in Data I/O's ABEL™ software to design GAL® devices. With Boolean equations, designers have total control over the logic within the pLSI or ispLSI devices. Also, complete access to the advanced architectural features such as the product term Clocks and Reset, the Output Enable control, the hardware XOR is provided.

As powerful as Boolean Equations are, it is time consuming to enter a large design using them. For that reason pDS software comes complete with a Macro library of standard logic functions which designers can draw from. The Macro library consists of several hundred logic elements ranging from simple gates (AND, OR, XOR) to complex functions like counters, multiplexers and adders. If a standard Lattice library Macro is close to design requirements, it can be copied to a personal library and modified. This new Macro is then saved and used in other designs when needed.

8

LATTICE SEMICONDUCTOR CORP., 5555 Northeast Moore Ct., Hillsboro, Oregon 97124, U.S.A.
Tel. 1-800-LATTICE (528-8423); FAX (503) 681-3037

For a non-standard logic function used repeatedly in a design, a Macro can be created using a combination of Boolean Equations and other Macros as described above.

Design Process

The design process in figure 8-17 includes the following simple steps:

1 Enter the Design
2 Verify the Logic
3 Route the Cells
4 Generate the Fusemap
5 Program the Part

Enter the Design

Entering the design is done using the graphical interface. The Lattice pDS software displays a block diagram of the part similar to the one shown in the data sheet. The design equations or Macros are entered by clicking on one of the Logic or I/O Cells using the Mouse, and writing the equations into the cell using a simple text editor. This editor is similar to the Windows Notepad. The graphical interface also allows advanced functions such as clearing a cell, naming a cell, copying the contents of one cell to another or saving the data in a cell to be recalled later.

Verify the Logic

Verifying the logic is done in two places. Each GLB and I/O Cell is verified individually. A Cell Verify is a local verify of that single cell only. It checks for problems such as syntax errors, exceeding the number of allowable cell inputs, outputs or product terms, and logic errors. Once the design is completely entered, the next step is to perform a Design Verify. The Design Verify performs a Cell Verify on cells which were not previously checked, and then checks all the interconnections within the device for dangling inputs and unconnected outputs. The design must pass a Design Verify before the following steps are performed.

Route the Cells

Routing the cells is the next step. The Place and Route module moves the GLBs and I/O Cells around in such a way that all of the networks which you have specified can be interconnected. If you have connected certain signals to specific pins, this information is entered into the design using a menu option in the Place and Route module. Aside from fixing the I/O pins, this is an entirely automatic process and requires no intervention. Due to the optimized design of the Global Routing Pool, place and route times are very fast, averaging a few minutes on most PCs.

Generate the Fusemap

The Fusemap generation module uses the routed design to generate the JEDEC file. The JEDEC file provides the data used to program the part. This file has a suffix of .JED. Like the Place and Route program, this is an automatic process.

Program the Part

The part is programmed in one of two ways. When using an external serial device programmer, the user can invoke a communication Program to transmit the JEDEC file to the programmer. When using in-system programming (isp) in a design, the Lattice design system invokes its own isp control program. This program uses a cable connected to the PC's Parallel Port to program the part or multiple parts on the board itself.

Clock Design Description

The clock design includes the following modules:

- ❑ Control Logic
- ❑ Prescaler
- ❑ Counters
  - Seconds
  - Minutes
  - Hours

Figure 8-18 shows a block diagram of the clock modules.

Control Logic

The Control Logic reads the input switches and controls the speed at which the seconds, minutes, and hours are incremented. This allows a user to set the clock.

Figure 8-17. Design Process

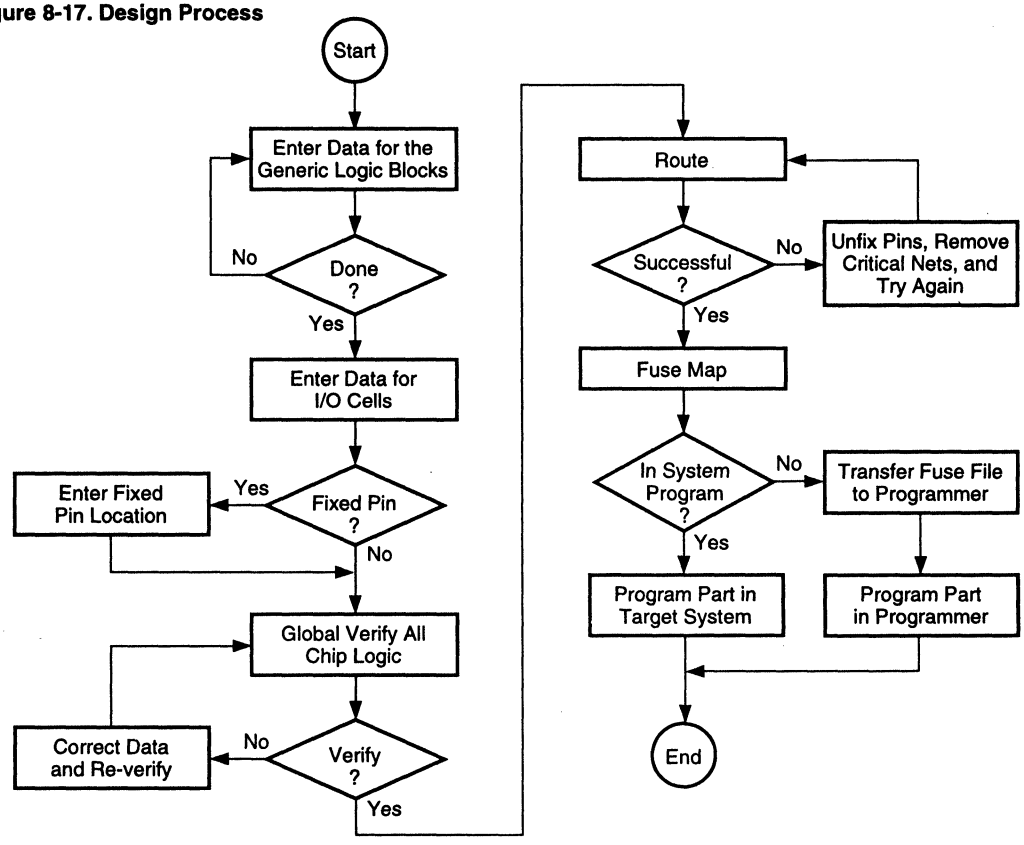
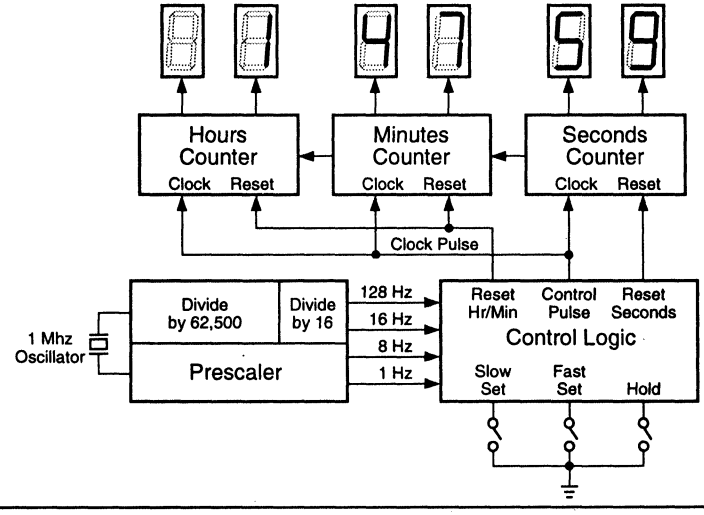


Figure 8-18. Block Diagram of the Clock



Slow Set, Fast Set, and Hold Signals

The Control Logic, shown in figure 8-19, generates the signals necessary to set and run the clock. The inputs to the Control Logic are the three switches: Slow Set, Fast Set and Hold. These inputs are clocked using the Input Register in the I/O Cells. This eliminates switch bounce which affects how the logic operates.

Timing Signals

The other signals coming into the control logic are the timing signals 128 Hz, 8 Hz and 1 PPM. These come from the prescaler and are used for the Fast Set function, Slow Set function and normal operating function respectively. The hours and minutes counters are normally clocked at a rate of 1 Pulse per Minute (1PPM). When you are setting the clock, this frequency is increased to 8 Hz for Slow Set and 128 Hz for Fast Set.

Figure 8-19 shows the schematic for this circuit and table 8-4 shows the Truth Table.

The Prescaler

The Prescaler divides the 1MHz clock, generated on the demo board, into the frequencies needed by the clock. The Prescaler is designed entirely using Macros from the Lattice library. The prescaler circuit has two purposes. It divides the 1MHz XTAL Oscillator signal down to 1 Hz for the seconds counter clock and also provides the frequencies necessary for the Slow Set and Fast Set functions.

Figure 8-19. Timing Signal Schematic

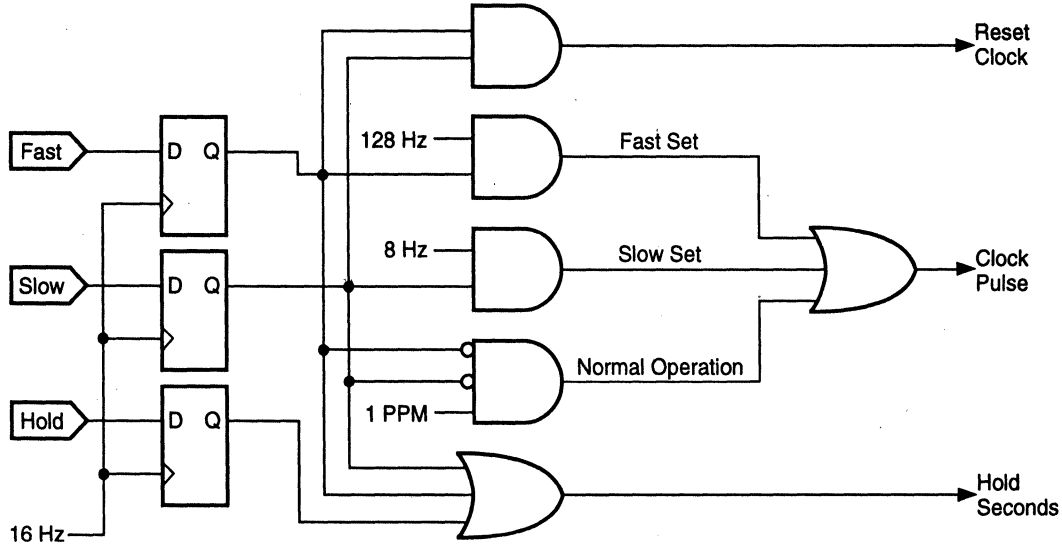


Table 8-4. Control Logic Truth Table

Slow	Fast	Hold	Clock Pulse	RST SEC	RST HR_MIN	Operation
0	0	0	1 Pulse Per Minute	0	0	Normal Operation
1	0	0	8 Hz	1	0	Slow Set
0	1	0	128 Hz	1	0	Fast Set
1	1	0	None	1	1	Reset to 12:00 AM
X	X	1	None	1	0	Hold Time

The circuit is implemented using two standard Macros from the Lattice library (see figure 8-21).

A 20-bit divider is necessary to divide the 1MHz clock signal down to 1 Hz, but the largest counter in the library is 8-bits. Therefore, three counter stages are needed to complete the division.

The approach chosen was to use two 8-bit preloadable counters and a 4-bit binary counter cascaded together. The two 8-bit counters are configured as a single 16-bit divider in this circuit. Because a binary counter was chosen for the 4-bit function, the mathematics are as follows:

1,000,000 Hz Divided by 16 = 62500 Hz.

Therefore, the output required of the 16-bit divider is 62500 Hz.

65535	Minus	62500	=	3035
Maximum count of the 16 bit counter	Minus	The desired Division	=	Preload Value

A 16-bit divider preloaded to 3035 (0BDB in Hexadecimal) at each terminal count has an output frequency of 16 Hz.

The frequencies necessary for the clock set functions are then chosen from the counter outputs. The 8 Hz signal (CBU14, Output Q0) advances the minutes counter at the rate of 1 minute every 7.5 Seconds. This is acceptable for the Slow Set function. The Fast Set function uses a 128Hz signal (C16Up, Output Q12) to advance the clock at a rate of 1 Hour every 2 seconds. The 16 Hz signal is used in the I/O cell input registers as a debounce clock for the switches.

In the final design, the 16-bit counter is placed in GLBs A0 through A7, and the 4-bit counter in GLBs B0 and B1.

Counters

The Seconds and Minutes Counters are Modulo 60 counters which display a decimal count ranging from 00 to 59.

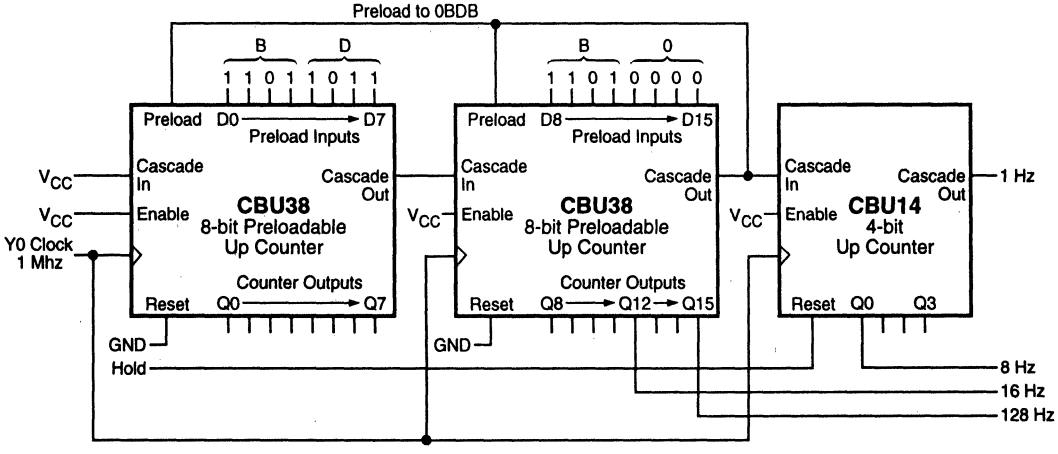
The Hours counter is a special Modulo 24 counter which counts from 1 to 12, and has a separate output bit for AM-PM indication. This counter resets to 12 AM and never displays a count of 00.

The Seconds counter is designed by using a standard Macro from the Lattice library and modifying it to suit the needs of the design. This combines the use of Macros with the use of Boolean equations.

The Minutes and Hours counters are designed using state machines optimized for the pLSI 1032 and ispLSI 1032 architectures. The counter which is created is then saved as a custom Macro for later use. This optimization saves time and effort on future designs.

There are three controls for setting the clock. These are Slow Set, Fast Set, and Hold. The Slow Set button advances the clock at a rate suitable for selecting the correct minute. The Fast Set button advances the clock at a rate suitable for selecting the correct hour. When either of these buttons are pressed, the seconds counters are reset to 00.

Figure 8-21. Prescaler Sample with Standard Macros



When Slow Set and Fast Set are pressed at the same time, the clock resets to 12:00 A.M.. The Hold button disables the minutes and hours counters from counting, and resets the seconds to 00 and holds that count until the button is released. This allows the clock to be set to the exact second.

The outputs from the circuit are the seven segment outputs from the Hours, Minutes and Seconds counters, and an AM/PM Indicator LED. Because the design must fit into the Demo board which was designed previously, the signals are fixed to specific device pins.

Seconds Counter

The Seconds counter is implemented using both a standard Macro from the library for the ones-of-seconds, and a modified counter Macro for the tens-of-seconds. The outputs of these counters is sent to two BCD and then to Seven Segment Display Macros to drive the LEDs.

The seconds counter counts from 0 to 59, and then resets to 0. An unmodified CDU24 decimal up counter is used for the Least Significant Digit, but the Most Significant Digit is a modulo 6 counter. This is not a standard function in the library. The easiest way to implement this function is to select a standard 4-bit binary counter (CDU24) and modify as shown in listing 3.

Listing 3.

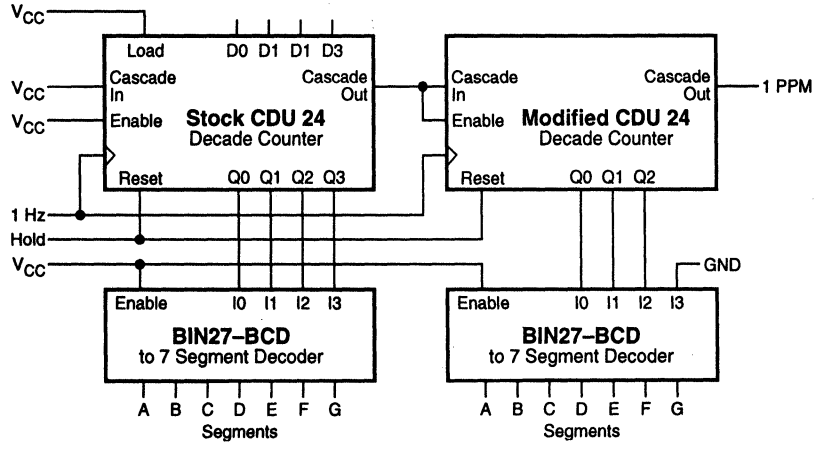
```
MACRO  MODULO6  ([Q0..Q2],CLK,EN,CS);

MACROTYPE  RX;
     MACROCOMMENT Custom 3 bit Modulo 6
     counter with Sync clear and enable
          for clock design;
     SIGTYPE [Q0..Q2] REG OUT;
     EQUATIONS
          Q0.CLK = CLK;
          Q0 = (Q0&!EN&!CS)
               $$ (!Q0&EN&!CS);
// Output Q0 toggles after counts
// 0,2,and 4.
          Q1 = (Q1&!EN&!CS)
          $$ ((!Q2&!Q1&Q0&EN&!CS)
          #  (!Q2&Q1&!Q0&EN&!CS));
// Output Q1 toggles after counts 1
// and 2.
          Q2 = (Q2&!EN&!CS)
          $$ ((!Q2& Q1& Q0&EN&!CS)
          #  (Q2&!Q1&!Q0&EN&!CS));
// Output Q2 toggles after counts 3
// and 4.
     END
END
```

This counter can then be saved in a personal library for future use.

The synchronous reset inputs to the seconds counters are driven by the Hold signal from the control logic. The clock is set to the exact second by setting the Hours and Minutes counters to a point several minutes ahead, and then pressing the Hold button until the correct second arrives (see figure 8-22).

The counters and the seven segment decoders were placed in GLBs, B2 through B7.

Figure 8-22. Sample Seconds Counter



The Prototyping Board

The ispLSI 1032 Prototyping Board was used to demonstrate the clock design. It is a simple design incorporating:

- An ispLSI 1032 device with all of the functional pins brought out to a connector.
- A 5 volt regulator requiring 9 to 15 Volts A.C. or D.C. as input. A calculator type power supply is provided with the board.
- Eight Discrete LEDs connected to eight of the I/O pins.
- An Eight Position DIP Switch connected to eight different I/O Pins.
- A Reset Switch with integral Power On LED.
- A 1MHz Crystal Oscillator providing a stable Clock Source.
- An in-system programming Cable connector (RJ45) for Programming.
- A "Breadboard" area for Wire Wrap sockets.
- A Daughter Board with eight 7 Segment Displays driven directly from 64 of the I/O Pins.

Figure : ispLSI Prototyping Board

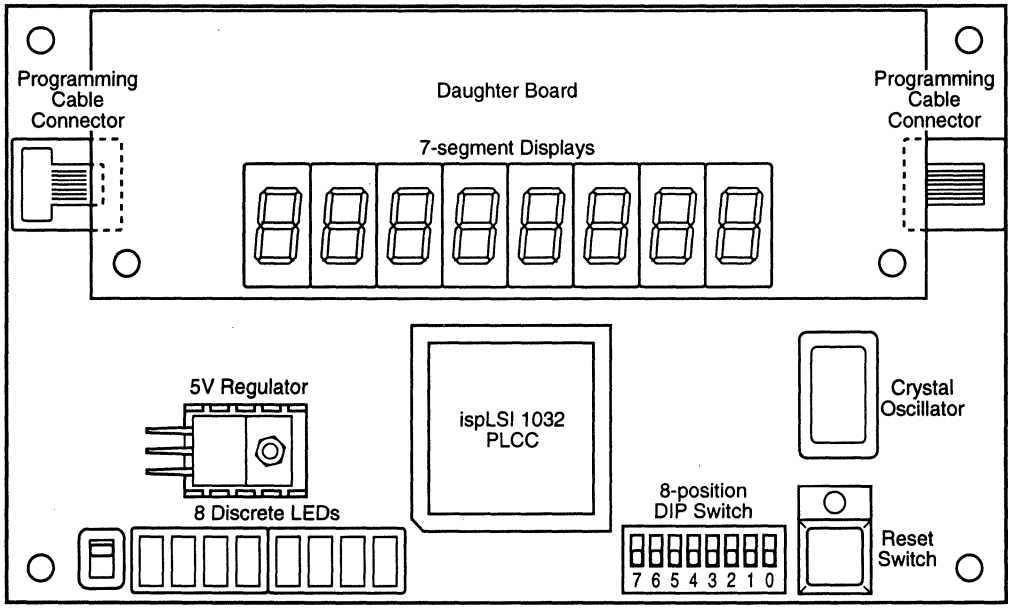


Table: Prototyping Board Pinouts

I/O #	Pin #	LED	Segment	Function	I/O #	Pin #	LED	Segment	Function
0	26	1	A	Not used	32	68	5	A	Minutes
1	27	1	B	Not used	33	69	5	A	Minutes
2	28	1	C	Not used	34	70	5	A	Minutes
3	29	1	D	Not used	35	71	5	A	Minutes
4	30	1	E	Not used	36	72	5	A	Minutes
5	31	1	F	Not used	37	73	5	A	Minutes
6	32	1	G	Not used	38	74	5	A	Minutes
7	33	1	DP	Not used	39	75	5	A	Minutes
8	34	2	A	Not used	40	76	6	A	10s of Minutes
9	35	2	B	Not used	41	77	6	A	10s of Minutes
10	36	2	C	Not used	42	78	6	A	10s of Minutes
11	37	2	D	Not used	43	79	6	A	10s of Minutes
12	38	2	E	Not used	44	80	6	A	10s of Minutes
13	39	2	F	Not used	45	81	6	A	10s of Minutes
14	40	2	G	Not used	46	82	6	A	10s of Minutes
15	41	2	DP	Not used	47	83	6	A	10s of Minutes
16	45	3	A	Not used	48	3	7	A	Hours
17	46	3	B	Not used	49	4	7	A	Hours
18	47	3	C	Not used	50	5	7	A	Hours
19	48	3	D	Not used	51	6	7	A	Hours
20	49	3	R	Not used	52	7	7	A	Hours
21	50	3	F	Not used	53	8	7	A	Hours
22	51	3	G	Not used	54	9	7	A	Hours
23	52	3	DP	Not used	55	10	7	A	Hours
24	53	4	A	Not used	56	11	8	A	10s of Minutes
25	54	4	B	Not used	57	12	8	A	10s of Minutes
26	55	4	C	Not used	58	13	8	A	Not used
27	56	4	D	Not used	59	14	8	A	Not used
28	57	4	E	Not used	60	15	8	A	Not used
29	58	4	F	Not used	61	16	8	A	Not used
30	59	4	G	Not used	62	17	8	A	Not used
31	60	4	DP	Not used	63	18	8	A	AM-PM Indicator

A modulo 6 counter is needed for the tens-of-seconds, and it is easily created by modifying a standard Modulo 10 Counter Macro. Once that new Macro is created, it is named and saved in the personal library.

The Minutes Counter

The architecture of the Lattice pLSI and ispLSI devices has been optimized for state machine use. The registers in the GLBs are synchronous and several product terms per register are added. Each product term has 18 inputs.

In the seconds counter, since the counters and the decoders are separate, seven GLBs are used. Taking advantage of the wide input gating available to create a state machine counter which directly drives the seven segment outputs, then the number of GLBs is reduced to four. Figure 8-23 shows a sample minutes counter.

The truth table for a seven segment display is shown in figure 8-24.

The state machine is simple. The outputs are the segment drivers, and each output decodes the current state to determine what the next state is. The simplified equation for segment A is shown in listing 4.

Listing 4. Segment A Equations

```
seg_A =  seg_A & seg_B & seg_C & seg_D
         & seg_E & seg_F & !seg_G
         // Decode state Zero
         # seg_A & seg_B & seg_C & seg_D
         & !seg_E & !seg_F & seg_G
         // Decode state Three
         # seg_A & !seg_B & seg_C
         & seg_D & !seg_E & seg_F
         & seg_G
         // Decode state Five
```

The output for segment A goes to zero on the following clock whenever states Zero, Three or Five occur. For each of the segments there are fewer zero transitions than one. The zero transitions are decoded to save product terms, and then inverted in the output buffers. This is true on all of the segments except Segment G, which is left in its logic true form. This allows the counter to reset to a Zero when a hardware reset is applied. All segments are on except segment G.

Figure 8-23. Sample Minutes Counter

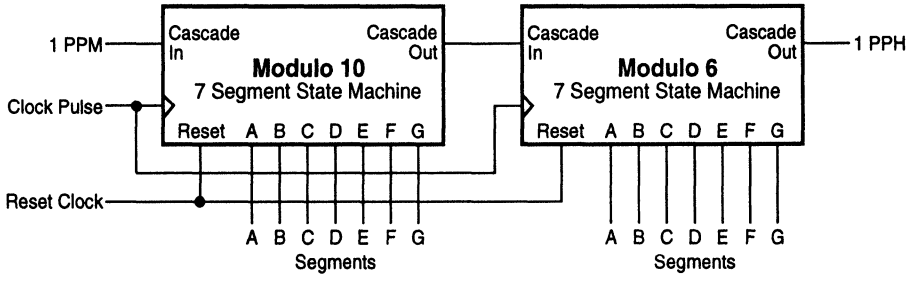
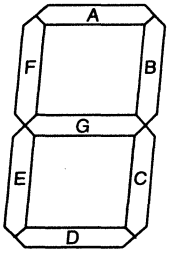


Figure 8-24. Seven Segment Truth Table



STATE	SEGMENT A	B	C	D	E	F	G	TC
0	1	1	1	1	1	1	0	0
1	0	1	1	0	0	0	0	0
2	1	1	0	1	1	0	1	0
3	1	1	1	1	0	0	1	0
4	0	1	1	0	0	1	1	0
5	1	0	1	1	0	1	1	0
6	0	0	1	1	1	1	1	0
7	1	1	1	0	0	0	0	0
8	1	1	1	1	1	1	1	0
9	1	1	1	0	0	1	1	1

The Terminal Count (TC) output enables the next stage. The tens-of-minutes counter is similar in construction except that only the states from zero to five are decoded, and the terminal count occurs at state five instead of nine (see figure 8-25).

By designing the counters to make best use of the features of the pLSI device family, logic for this counter function is reduced by 40%. The minutes counters are placed in GLBs C0 through C7 in the final design.

The Hours Counter

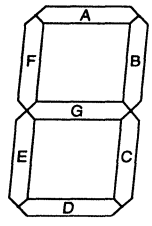
The hours counter is constructed using a state machine similar to the one used in the minutes counter. The count sequence for hours is unique compared to most counters.

In the hours stage, both digits are designed as a single counter stage. The reset signal for the hours stage resets the counter to 12 rather than zero (see figure 8-26).

A carry out signal is still generated from this counter because an AM/PM indicator is desired, but the carry out is generated when the counter reaches 12 instead of when it rolls over to one. This is consistent with the way clocks operate. Morning starts at 12:00 AM and afternoon starts at 12:00 PM. The AM/PM stage is a D-type flip-flop which uses the carry out signal as an asynchronous product term clock. This register also uses an asynchronous reset to force it to start at 12:00 AM when the clock is reset (see figure 8-27).

The hours counter and the AM/PM logic are placed in GLBs D0 through D4 in the example file.

Figure 8-25. Terminal Count at State 5



STATE	SEGMENT A	B	C	D	E	F	G	TC
0	1	1	1	1	1	0	0	0
1	0	1	1	0	0	0	0	0
2	1	1	0	1	0	1	1	0
3	1	1	1	1	0	1	1	0
4	0	1	1	0	1	1	1	0
5	1	0	1	1	1	1	1	1

Figure 8-26 Sample Hours Counter

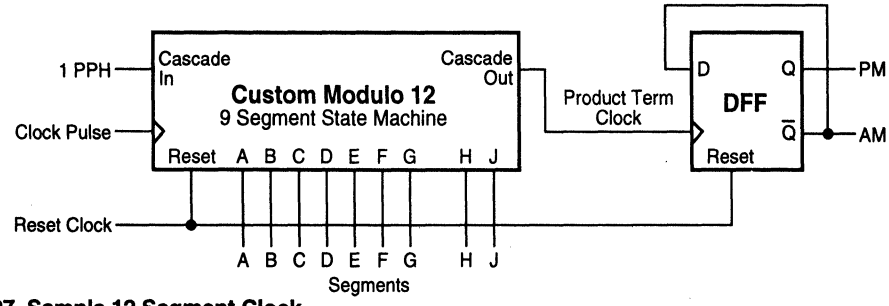
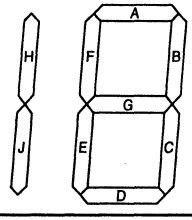


Figure 8-27. Sample 12 Segment Clock



STATE	SEGMENT A	B	C	D	E	F	G	H	J	TC
1	0	1	1	0	0	0	0	0	0	1
2	1	1	0	1	1	0	1	0	0	1
3	1	1	1	1	0	0	1	0	0	1
4	0	1	1	0	0	1	1	0	0	1
5	1	0	1	1	0	1	1	0	0	1
6	0	0	1	1	1	1	1	0	0	1
7	1	1	1	0	0	0	0	0	0	1
8	1	1	1	1	1	1	1	0	0	1
9	1	1	1	0	0	1	1	0	0	1
10	1	1	1	1	1	1	0	1	1	1
11	0	1	1	0	0	0	0	1	1	0
12	1	1	0	1	1	0	1	1	1	1

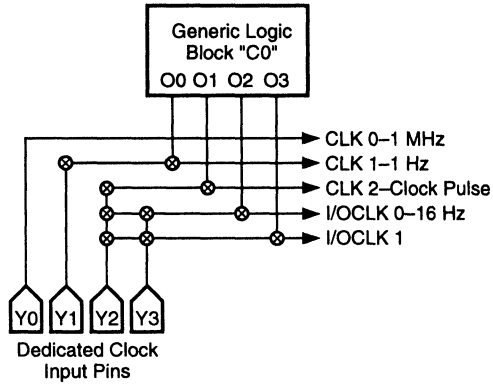
Clock Management

This design makes maximum use of the various Clock modes of the pLSI and ispLSI Family. In each GLB, there are four possible clock sources, CLK 0, CLK 1, CLK 2, and a PTCLK. The first source, CLK 0, is a synchronous clock, and is permanently connected to the Y0 Clock Input pin on the device. CLK 1 and CLK 2 are also synchronous and can come from either the external clock pins (Y1 or Y2) or can be generated within the device using the internal clock GLB, "C0". The fourth, PTCLK, comes from a Product Term within the GLB. This clock is asynchronous (see figure 8-28).

In this clock design, the 1MHz reference clock from the Demo Board is brought in using the Y0 Clock pin. It is the clock source used to drive the Prescaler. The 1 Hz output of the Prescaler is then routed through the "C0" GLB to become the CLK 1 Source. This clock is used to increment the seconds counters. The minutes and hours counters are clocked by the signal Clock Pulse on the CLK 2 distribution line. This signal is 1 Pulse per Minute during normal operation, 8 Hz during Slow Set and 128 Hz during Fast Set Operations.

The AM/PM Indicator is a D-type flip-flop which is clocked asynchronously using a product term clock (see figure 8-27).

Figure 8-28. Clock Management Modes



in-system programming

The Lattice ispLSI (in-system programmable Large Scale Integration) devices are programmable in circuit on a powered board. A special programming cable plugs into a parallel printer port on a PC to support in-system programming.

in-system programming (isp) is a powerful feature. It simplifies the design flow by eliminating the time consuming simulation process. The design can be tested in the final system by downloading the JEDEC file directly into the part. This is especially useful in surface mount environments where the parts cannot be removed from the board for programming. Test points are brought out to unused I/O pins during the debug cycle, and eliminated for standard operation. A designer can complete the design in steps by creating smaller modules of the design, testing them as stand alone circuits, and then combining them once they are all working correctly.

In addition to being a design tool, in-system programming also offers production advantages. Field Service upgrades are performed by simply reprogramming the boards, and options are added by programming them into the logic. If several boards are similar in function, but have different logic, a single printed circuit board is designed, and the function is programmed into the logic just before the board is shipped. This reduces both production and inventory costs.

The only requirements of the system are that it must have a stable 5 volt V_{CC}, and have a connection point for the isp cable. The standard interface used on the ispLSI prototyping boards is a common 8-pin telephone connector. This connector is selected because it is small, reliable and inexpensive. See the figure showing isp Cable.

Five pins on the ispLSI 1032 device are dedicated to programming when the part is used in the isp mode. They are:

$\overline{\text{ispEN}}$	in-system programming Enable
MODE	isp Mode Control
SCLK	Shift Clock
SDI	Serial Data In
SDO	Serial Data Out

in-system programming (Continued)

In addition to these signals, Power, Ground and Reset leads are provided in the cable. Power and Ground are supplied by the target system and are used by logic in the isp cable. Reset is an optional signal connected to System Reset in the target system.

Once the JEDEC (.JED) file is created, a part can be programmed. To do this, ensure that the cable is connected to the target system and that the 5 Volts VCC is on. From the Design menu, select the program option to access the Lattice isp Programming module. The isp program first calibrates itself to the speed of the computer. It then prompts for the printer port the isp cable is connected to, LPT1, LPT2, or LPT3. Once communication has been established with the ispLSI part to be programmed, the program prompts for the name of the .JED file used for programming the part.

The algorithm which is used to program the part is straightforward. The MODE, SCLK and SDI pins are used to control a state machine internal to the ispLSI device. The device is controlled by serially shifting a series of commands and data streams. The State diagram for that operation is shown in the isp State Machine figure.

Figure: isp Cable with RJ45 Connector

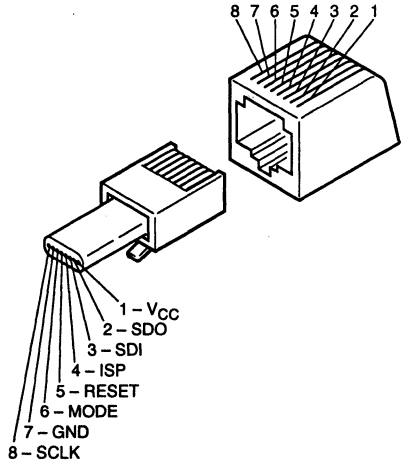
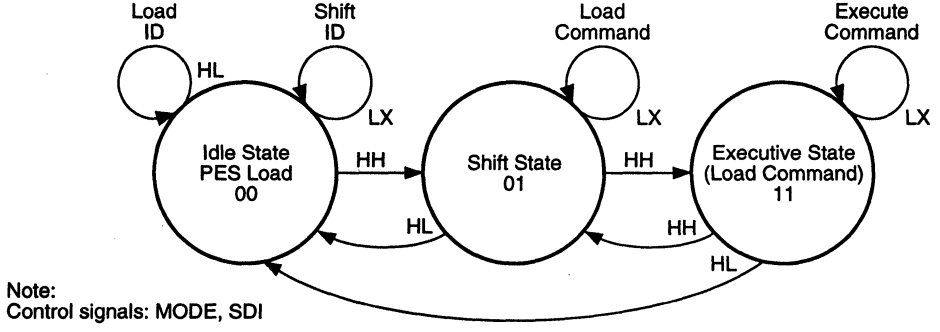


Figure: isp State Machine



Note:
Control signals: MODE, SDI

ispLSI Configurable Memory Controller

January 1992 **Application Note 2502**

Introduction

There are many advantages of using the in-system programmable ispLSI devices. In board level designs, as well as during manufacturing, the flexibility of hardware reconfiguration can lead to many innovative system designs. Once configured the ispLSI devices' non-volatile E^2CMOS cells will retain the configuration even when the power is turned off. The guaranteed 1,000 programming cycles and the 20 year data retention of the ispLSI device will allow the user to reliably reconfigure the device as often as required.

This application note highlights the advantages of having the flexibility when designing with the ispLSI device and how this can lead to innovative design ideas which translate to ease of use and instant updates without board layout change. The flexibility of design is illustrated with the use of the Dynamic Random Access Memory (DRAM) controller as an example. The example shows a typical microprocessor and memory interface with the memory controller controlling the DRAM access and refresh timing requirements. The use of Lattice pLSI/ispLSI Development System (pDS) software is also illustrated in this application note. The Lattice Design File (.ldf) listing and the report files generated by the software are also attached at the end of this section.

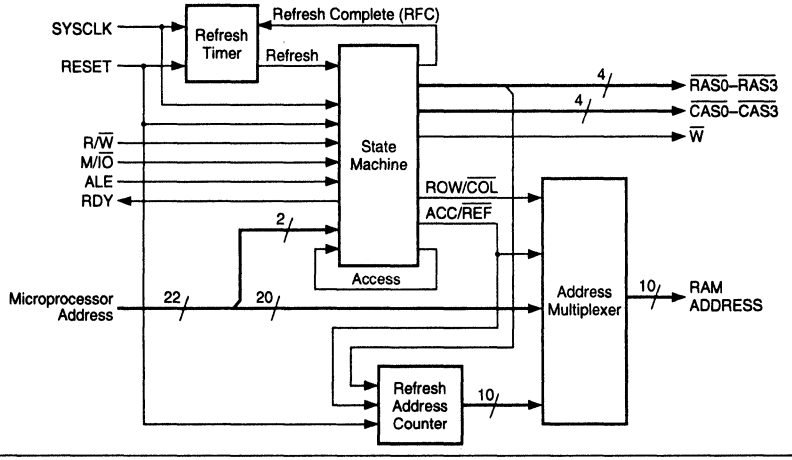
Memory Controller Logic Overview

When interfacing the microprocessor to the DRAM, the control signal and timing requirements of both the processor and the DRAM must be satisfied. In order to satisfy these requirements, the external timing controller must take the processor address, data and control signals and translate them into the control signals for the DRAM. At the same time, the DRAM timing controller must take into account the refresh requirements of the DRAM.

Figure 8-28 shows the block diagram of the DRAM timing controller that is implemented in the ispLSI 1032. The state machine and address multiplexer blocks are used to control the memory access request of the processor and supply the DRAM with the necessary address and control signals. DRAM refresh requirements are controlled by the refresh timer block, refresh address counter block and the address multiplexer block.

Any access request from the processor is processed by the state machine based on the processor control signals such as Read/Write (R/W), Memory/IO access (M/IO), Address Latch Enable (ALE) and the microprocessor address signals. The Ready (RDY) signal is used to inform the processor the status of the requested data. In other words,

Figure 8-28. DRAM Timing Controller Block Diagram



LATTICE SEMICONDUCTOR CORP., 5555 Northeast Moore Ct., Hillsboro, Oregon 97124, U.S.A.
Tel. 1-800-LATTICE (528-8423); FAX (503) 681-3037

it is used to acknowledge the processor that the memory is ready to respond to the processor. The address multiplexer generates the row and column addresses necessary for the memory access cycle. The appropriate Row Address Strobe (RAS), Column Address Strobe (CAS), and Write (W) signals are also generated by the state machine based on the processor inputs. To arbitrate between memory access request and the refresh request, the state machine also generates the status signal called Access. The purpose of this signal is to keep track of an access cycle when the refresh sequence is in progress. This status signal is then used to determine whether or not to begin an access sequence after the refresh sequence. As part of the access/refresh arbitration, the state machine also issues an Access/Refresh (ACC/REF) signal to the address multiplexer logic block. Based on this signal the address multiplexer block routes the appropriate access or refresh address on to the external DRAM address bus.

As for all DRAMs, memory refresh must be completed within a specified time. This process is completely controlled by the DRAM timing controller. The refresh timer block generates the internal refresh request signal according to the system clock speed and the DRAM refresh rate requirements. When the state machine detects this refresh request signal, the refresh sequence for the DRAM is generated as soon as time permits. This means that the refresh sequence is generated right after the refresh request or if the timing controller is in the middle of a memory access cycle the refresh sequence is generated right after the memory access cycle is complete. During the refresh sequence the row address and all the RAS signal must be activated to perform the basic RAS-only refresh. The row addresses are supplied by the refresh address counter logic block. This logic block keeps track of the rows that are being refreshed and it gets incremented every time a refresh sequence is performed. All the RAS signal are activated for refresh by the state machine.

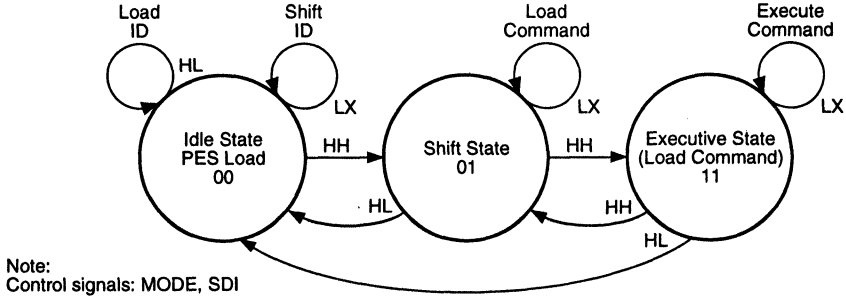
With the basic understanding of the DRAM timing control logic complete, the next section will discuss the implementation of the logic in an ispLSI device and how to take advantage of the isp features to make the system design, manufacturing and field updates easy and flexible.

Taking Advantage of isp Features

Implementing a basic DRAM timing control logic in the ispLSI 1032 takes up approximately 65% of the total logic available in the device. (It is with this in mind that the features needed for a specific design can be added to these basic logic blocks). With the isp capability, many features can be added to accommodate the ever changing requirements of the system, microprocessor speeds, availability of DRAMs, and the memory configurations. Moreover, the changes are made only under the software control. Instead of having different production runs for various different options, the options are added at the in-system programming stage.

The programming of the ispLSI devices are handled via five TTL level interface signals. Of these five signals, four signals can be dual function, a programming function as well as an input during normal device operation. The isp Enable ($\overline{\text{ispEN}}$) signal is the one dedicated programming pin used to enable and disable the programming function. Once in programming mode, the mode control (MODE), serial data input (SDI), serial data clock (SCLK), and serial data output (SDO) signals control the entire programming process. The address and data required to program the device are serially shifted into the internal shift registers and the three state programming state machine steps through the programming sequence. The five-bit instructions within the state machine define all the necessary steps for programming. Figure 8-29 shows the isp programming state machine with the control signal requirements for the state transitions. Refer to the isp programming section 4 of the data book for a more detailed programming description.

Figure 8-29. isp State Machine



Different System Speed

Designing with a different speed microprocessor requires a different DRAM timing controller. The adjustments must be made in the state machine and refresh timer logic of the controller to account for the difference in speed. Without the capabilities of the isp features, different boards with different PLD codes must be built to work with different processor speeds. By providing a simple programming circuitry on board to support the isp programming, the logic adjustments for different speed processor can be accomplished by in-system programming the different patterns via software control. Manufacture of these options are made simple and cost effective by not having to keep an inventory of prepatterned devices.

DRAM Feature Flexibility

DRAMs have many features from which the system designer can select. For the same DRAM configuration, the system designer can select from DRAMs that have different access schemes such as nibble mode, static column mode and page mode. Similarly, different memory refresh schemes can be chosen. The two choices of refresh schemes include the simple RAS only refresh and the option to perform hidden refresh with the CAS before RAS refresh scheme. Most of these various DRAM options can be supported by in-system programming the ispLSI devices. Again, the flexibility lies in the fact that the decision of what function the ispLSI will perform on board can be made after the decision has been made on which type of DRAMs are used on board.

Different DRAM Configuration

The ispLSI implementation of the DRAM timing controller makes the change of memory configuration very simple. Reprogramming of the address decoding and turning on the appropriate address strobe signals for different memory configuration can be done by in-system reconfiguration of the state machine and the address decoding of the ispLSI device. All of these changes can be accomplished under software control.

Memory Timing Controller Details

As shown in figure 1 the memory timing controller consists of four different logic blocks. The refresh timer, state machine, refresh address counter and memory address multiplexer. All boolean equations for the logic blocks are developed within the Lattice pDS software. The entire memory timing controller design assumes that all the processor signals are typical of a commercially available processor with a clock speed of 25 MHz. DRAMs are arranged in four banks of 1M X 32-bit arrangement. All timing for the access and refresh sequences are shown in the timing diagram.

Refresh Timer

The function of the refresh timer is to generate a refresh request signal every 15.5 µs. This refresh period is derived from the DRAM refresh requirement of 512 rows of refresh every 8 ms for the 1M X 1 DRAM. Based on the 25 MHz system clock frequency, the count value to divide the clock period to the refresh period is 200. Changing processor speed will only require a change of count value. Once the count value expires, the refresh timer generates an internal refresh signal to inform the state machine to perform a refresh cycle. When the state machine completes the refresh cycle, a refresh complete (RFC) signal is generated for the refresh timer. The refresh timer then resets the internal counter for the next refresh period.

ispLSI implementation of the refresh timer takes up three GLBs (A0-A2) within the device. The system clock is used to run the nine bit counter, RFC is the input signal to this block and REFRESH is the output signal of this logic block.

State Machine

The state machine can be further divided into four different sub-logic blocks. These sub-logic blocks consists of a RAS generator, CAS generator, 4-bit state machine which is divided into two state variable bits and two counter bits, and control signal generator. In the ispLSI 1032 implementation, the state machine logic block takes up 9 GLBs.

The 4-bit state machine is divided into a 2-bit state variable, named ST0 and ST1, and 2-bit state counter, named SCNT0 and SCNT1. The state diagram with its state transitions are shown in figure 8-30. In each of the access and refresh states, the state counter sequences through the operation until the sequence is complete. The purpose of the state variable bits are only to keep track of the state transitions. Once the state transition has occurred, the state counter bits take the responsibility of sequencing through the state.

The three states are divided as idle state, access state and refresh state. Based on the processor control signal and the internal refresh request signal, the state transition occurs from idle state to either access state or refresh state. If the refresh and access request happen at the same time, refresh request takes precedence over access request. When the refresh request is asserted during an access cycle, the refresh cycle follows right after the access cycle. The only other condition between the access and refresh request that the state machine needs

to arbitrate is when the access request occurs during the refresh sequence. The access feedback signal of the state machine is activated when the access request occurs during the refresh cycle. When the refresh cycle is complete, the access feedback signal is used to determine whether or not the access sequence needs to begin. The timing diagrams in figure 8-31 and 8-32 illustrate the control signal sequence for the access and refresh cycles, respectively.

In addition to the external DRAM control signals, the state machine also generates the control signal for the address multiplexer and the refresh address counter. The ROW/COL signal directs the address multiplexer to output the appropriate row and column address during the access cycle. Furthermore, the address multiplexer accepts the access/refresh (ACC/REF) control signal to either direct the memory access address from the processor, or direct the refresh row address from the refresh address counter to the DRAM.

Figure 8-30. DRAM Timing Controller State Machine

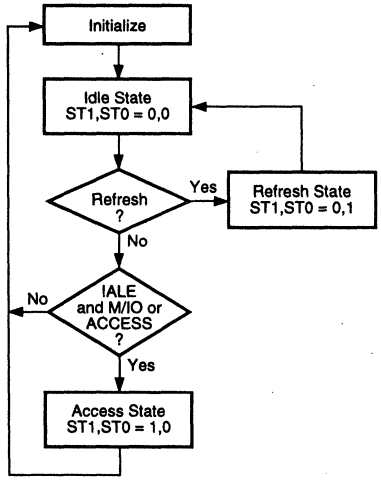
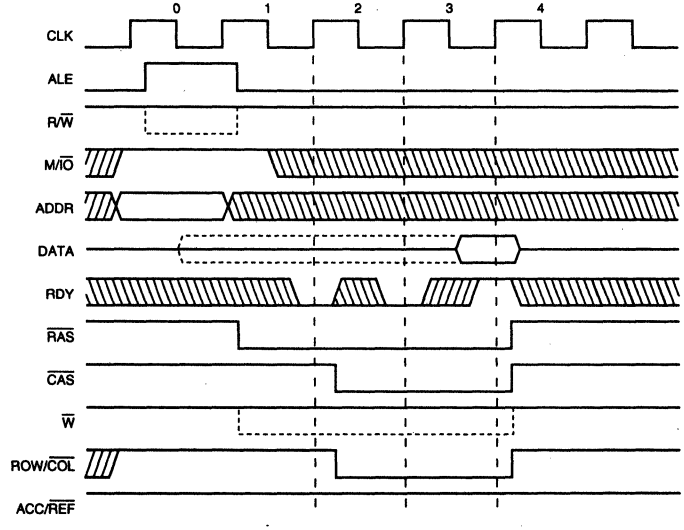


Figure 8-31. Access Cycle Timing



Refresh Address Counter

The refresh address counter keeps track of the rows of DRAM to be refreshed. This counter is only incremented on the falling edge of the RAS signal during refresh sequence. The ispLSI device implementation of this counter takes 3 GLBs.

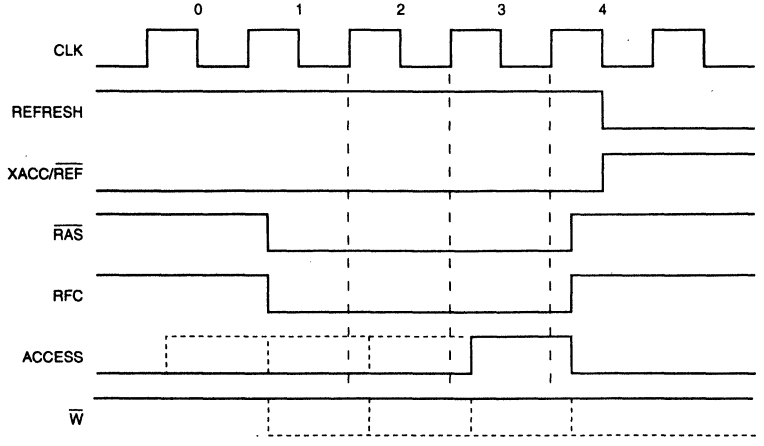
Memory Address Multiplexer

In access mode, determined by the ACC/REF internal signal, the memory address multiplexer multiplexes between the row and column address. Once in the refresh cycle, the refresh address comes from the refresh address counter. It takes 3 GLBs to implement the multiplexer in the ispLSI 1032.

Conclusion

The intention of this application section is to give an overview of how the isp features can be used to improve the design features and the manufacturing process by using an example of a generalized DRAM timing controller. In addition, the software example given in the document should provide a good starting point for designers who need to implement a state machine based design. With the flexibility of the ispLSI devices the possibilities are limited only by one's imagination to implement innovative designs. The following sections lists the Lattice Design file with the Boolean Equations and the report files which lists the device utilization and the pinout for the ispLSI 1032.

Figure 8-32. Refresh Cycle Timing



Design LDF Listing

```
//isp_app.ldf  generated  using  Lattice  pLSI  V1.00.19

LDF  1.00.00  DESIGNLDF;
DESIGN  DRAM  CONTROLLER  1.00;
PROJECTNAME  isp  APPLICATIONS;
DESCRIPTION
DRAM  CONTROLLER  DESIGN  FOR  isp  APPLICATION.

IT  INCLUDES  FOUR  MAJOR  BLOCKS.
    - REFRESH TIMER
    - REFRESH ROW ADDRESS COUNTER
    - ADDRESS MUX
    - STATE MACHINE;

PART  pLSI  1032-80LJ;

DECLARE
END;   //DECLARE

SYM GLB   C2   1   ;
    /////      ROW ADDRESS STROBE (RAS1,RAS0) GLB        /////
    SIGTYPE  IRAS1  REG OUT;
    SIGTYPE  IRAS0  REG OUT;
    EQUATIONS
        IRAS1.CLK = ICLK;
        IRAS1 = !ST0 & !IA20 & IRAS1 & !IRESET       /////    REDUCED RAS1      /////
              # !ST1 & IA21 & IRAS1 & !IRESET
              # !ST0 &  ST1 & SCNT0 & SCNT1 & IA20 & !IA21 & !IRESET
              #  ST0 & !ST1 & SCNT0 & SCNT1 & !IRESET
              # !ST0 & !ST1 & IRAS1 & !IRESET
              #  ST0 & ST1 & IRAS1 & !IRESET
              #  SCNT1 & IRAS1 & !IRESET
              #  SCNT0 & IRAS1 & !IRESET;

        IRAS0 = !ST0 & IA20 & IRAS0 & !IRESET        /////    REDUCED RAS0      /////
              # !ST1 & IA21 & IRAS0 & !IRESET
              # !ST0 &  ST1 & SCNT0 & SCNT1 & !IA20 & !IA21 & !IRESET
              #  ST0 & !ST1 & SCNT0 & SCNT1 & !IRESET
              # !ST0 & !ST1 & IRAS0 & !IRESET
              #  ST0 & ST1 & IRAS2 & !IRESET
              #  SCNT1 & IRAS0 & !IRESET
              #  SCNT0 & IRAS0 & !IRESET;
   END
END;

SYM GLB   A2   1   ;
    /////      REFRESH TIMER GLB2       /////
    SIGTYPE  RQ8  REG OUT;
    SIGTYPE  RQ9  REG OUT;
    SIGTYPE  REFRESH  REG OUT;
    FJK11  (REFRESH,R_RATE,RFC,ICLK);   /////    REFRESH REQUEST SIGNAL    /////
    EQUATIONS
        RQ8.CLK = ICLK;
        RQ8 = (RQ8 & !RFC)
            $$ (RQ7 & RQ6 & RQ5 & RQ4 & RQ3 & RQ2 & RQ1 & RQ0 & !RFC);
```

```
        RQ9 = (RQ9 & !RFC)
            $$ (RQ8 & RQ7 & RQ6 & RQ5 & RQ4 & RQ3 & RQ2 & RQ1 & RQ0 & !RFC);
        R_RATE = RQ7 & RQ6 & !RQ5 & !RQ4 & RQ3 & !RQ2 & !RQ1 & !RQ0;
    END
END;

SYM GLB  A1  1  ;
    /////      REFRESH TIMER GLB1      /////
    SIGTYPE RQ4 REG OUT;
    SIGTYPE RQ5 REG OUT;
    SIGTYPE RQ6 REG OUT;
    SIGTYPE RQ7 REG OUT;
    EQUATIONS
        RQ4.CLK = ICLK;
        RQ4 = (RQ4 & !RFC)
            $$ (RQ3 & RQ2 & RQ1 & RQ0 & !RFC);
        RQ5 = (RQ5 & !RFC)
            $$ (RQ4 & RQ3 & RQ2 & RQ1 & RQ0 & !RFC);
        RQ6 = (RQ6 & !RFC)
            $$ (RQ5 & RQ4 & RQ3 & RQ2 & RQ1 & RQ0 & !RFC
        RQ7 = (RQ7 & !RFC)
            $$ (RQ6 & RQ5 & RQ4 & RQ3 & RQ2 & RQ1 & RQ0 & !RFC);
    END
END;

SYM GLB  A0  1  ;
    //////  REFRESH TIMER GLB0    /////
    SIGTYPE RQ0 REG OUT;
    SIGTYPE RQ1 REG OUT;
    SIGTYPE RQ2 REG OUT;
    SIGTYPE RQ3 REG OUT;
    EQUATIONS
        RQ0.CLK = ICLK;
        RQ0 = !RQ0 & !RFC;
        RQ1 = (RQ1 & !RFC)
            $$ (RQ0 & !RFC);
        RQ2 = (RQ2 & !RFC)
            $$ (RQ1 & RQ0 & !RFC);
        RQ3 = (RQ3 & !RFC)
            $$ (RQ2 & RQ1 & RQ0 & !RFC);
    END
END;

SYM GLB  D0  1  ;
    /////      ADDRESS MUX GLB0       /////
    SIGTYPE IRAM0 ASYNC OUT;
    SIGTYPE IRAM1 ASYNC OUT;
    SIGTYPE IRAM2 ASYNC OUT;
    SIGTYPE IRAM3 ASYNC OUT;
    EQUATIONS
        IRAM0 = ROW_COL & ACC_REF & IA0     /////  ROW SELECT    /////
            #  !ROW_COL & ACC_REF & IA10 /////  COLUMN SELECT    /////
            #  !ACC_REF & RCNTR0;                 /////   REFRESH ADDR SELECT  /////
        IRAM1 = ROW_COL & ACC_REF & IA1
            #  !ROW_COL & ACC_REF & IA11
            #  !ACC_REF & RCNTR1;
        IRAM2 = ROW_COL & ACC_REF & IA2
            #  !ROW_COL & ACC_REF & IA12
```

```
            #   !ACC_REF & RCNTR2;
        IRAM3 = ROW_COL & ACC_REF & IA3
            #   !ROW_COL & ACC_REF & IA13
            #   !ACC_REF & RCNTR3;
    END
END;

SYM GLB   D1   1   ;
    /////       ADDRESS MUX GLB1          /////
    SIGTYPE  IRAM4  ASYNC  OUT;
    SIGTYPE  IRAM5  ASYNC  OUT;
    SIGTYPE  IRAM6  ASYNC  OUT;
    SIGTYPE  IRAM7  ASYNC  OUT;
    EQUATIONS
        IRAM4 = ROW_COL & ACC_REF & IA4      /////   ROW SELECT     /////
            #   !ROW_COL & ACC_REF & IA14 /////   COLUMN SELECT     /////
            #   !ACC_REF & RCNTR4;                  /////    REFRESH ADDR SELECT   /////
        IRAM5 = ROW_COL & ACC_REF & IA5
            #   !ROW_COL & ACC_REF & IA15
            #   !ACC_REF & RCNTR5;
        IRAM6 = ROW_COL & ACC_REF & IA6
            #   !ROW_COL & ACC_REF & IA16
            #   !ACC_REF & RCNTR6;
        IRAM7 = ROW_COL & ACC_REF & IA7
            #   !ROW_COL & ACC_REF & IA17
            #   !ACC_REF & RCNTR7;
    END
END;

SYM GLB   D2   1   ;
    /////       ADDRESS MUX GLB2          /////
    SIGTYPE  IRAM8  ASYNC  OUT;
    SIGTYPE  IRAM9  ASYNC  OUT;
    EQUATIONS
        IRAM8 = ROW_COL & ACC_REF & IA8      /////   ROW SELECT     /////
            #   !ROW_COL & ACC_REF & IA18 /////   COLUMN SELECT     /////
            #   !ACC_REF & RCNTR8;                  /////    REFRESH ADDR SELECT   /////
        IRAM9 = ROW_COL & ACC_REF & IA9
            #   !ROW_COL & ACC_REF & IA19
            #   !ACC_REF & RCNTR9;
    END
END;

SYM GLB   D5   1   ;
//////   REFRESH ROW COUNTER GLB0      /////
    SIGTYPE  RCNTR0  REG  OUT;
    SIGTYPE  RCNTR1  REG  OUT;
    SIGTYPE  RCNTR2  REG  OUT;
    SIGTYPE  RCNTR3  REG  OUT;
    EQUATIONS
        RCNTR0.PTCLK = !IRAS0;        /////      USE RAS AS THE COUNTER CLOCK     ////
        RCNTR0 = !RCNTR0 & !ACC_REF      /////    COUNT DURING REFRESH         /////
              #    RCNTR0 & ACC_REF;          /////      HOLD DURING ACCESS             /////
        RCNTR1 = (RCNTR1 & !ACC_REF)
               $$ ((RCNTR0 & !ACC_REF)
              #    (RCNTR1 & ACC_REF));
        RCNTR2 = (RCNTR2 & !ACC_REF)
               $$ ((RCNTR1 & RCNTR0 & !ACC_REF)
              #    (RCNTR2 & ACC_REF));
        RCNTR3 = (RCNTR3 & !ACC_REF)
```

```
              $$ ((RCNTR2 & RCNTR1 & RCNTR0 & !ACC_REF)
             #    (RCNTR3 & ACC_REF));
   END
END;

SYM GLB  D6  1  ;
    //////  REFRESH ROW COUNTER GLB1    /////
    SIGTYPE RCNTR4 REG OUT;
    SIGTYPE RCNTR5 REG OUT;
    SIGTYPE RCNTR6 REG OUT;
    SIGTYPE RCNTR7 REG OUT;
    EQUATIONS
       /////     USE RAS AS THE COUNTER CLOCK    ////
       RCNTR4.PTCLK = !IRAS0;
       RCNTR4 =   (RCNTR4 & !ACC_REF)
       /////  COUNT DURING REFRESH    /////
              $$ ((RCNTR3 & RCNTR2 & RCNTR1 & RCNTR0 & !ACC_REF)
             #    (RCNTR4 & ACC_REF));
       /////     HOLD DURING ACCESS        /////
       RCNTR5 = (RCNTR5 & !ACC_REF)
              $$ ((RCNTR4 & RCNTR3 & RCNTR2 & RCNTR1 & RCNTR0 & !ACC_REF)
             #    (RCNTR5 & ACC_REF));
       RCNTR6 = (RCNTR6 & !ACC_REF)
              $$ ((RCNTR5 & RCNTR4 & RCNTR3 & RCNTR2 & RCNTR1 & RCNTR0 & !ACC_REF)
             #    (RCNTR6 & ACC_REF));
       RCNTR7 = (RCNTR7 & !ACC_REF)
              $$ ((RCNTR6 & RCNTR5 & RCNTR4 & RCNTR3 & RCNTR2 & RCNTR1 & RCNTR0 &
                  !ACC_REF)
             #    (RCNTR7 & ACC_REF));
   END
END;

 SYM GLB  D7  1  ;
 //////  REFRESH ROW COUNTER GLB2    /////
    SIGTYPE RCNTR8 REG OUT;
    SIGTYPE RCNTR9 REG OUT;
    EQUATIONS
       RCNTR8.PTCLK = !IRAS0;      /////     USE RAS AS THE COUNTER CLOCK    ////
       RCNTR8 = (RCNTR8 & !ACC_REF)
              $$ ((RCNTR7 & RCNTR6 & RCNTR5 & RCNTR4
      ,  ///// COUNT DURING REFRESH    /////
      & RCNTR3 & RCNTR2 & RCNTR1 & RCNTR0 & !ACC_REF) # (RCNTR8 & ACC_REF));
       /////    HOLD DURING ACCESS          /////
       RCNTR9 = (RCNTR9 & !ACC_REF)
              $$ ((RCNTR8 & RCNTR7 & RCNTR6 & RCNTR5 & RCNTR4 & RCNTR3 & RCNTR2 &
                   RCNTR1 & RCNTR0 & !ACC_REF)
              #    (RCNTR9 & ACC_REF));
   END
END;

SYM GLB  C7  1  ;
    /////      STATE BITS GLB       /////
    SIGTYPE ST0 REG OUT;
    SIGTYPE ST1 REG OUT;
     FJK11  (ST0,JST0,KST0,ICLK);
     FJK11  (ST1,JST1,KST1,ICLK);
    EQUATIONS
       JST0 = !ST1 & !ST0 & REFRESH;            /////    STATE BIT0 SET INPUT    /////
       KST0 = !ST1 & ST0 & SCNT1 & SCNT0;      /////     STATE BIT0 RESET INPUT    /////
```

```
        JST1 = !ST1 & !ST0 & !REFRESH & !IALE & IMIO_
             # !ST1 & !ST0 & !REFRESH & ACCESS;       /////    STATE BIT1 SET INPUT    /////
        KST1 = ST1 & !ST0 & SCNT1 & SCNT0
             # !ST1 & ST0 & SCNT1 & SCNT0;          /////    STATE BIT0 RESET INPUT    /////
    END
END;

SYM GLB  C6  1  ;
/////       STATE COUNTER BITS GLB        /////
    SIGTYPE SCNT0 REG OUT;
    SIGTYPE SCNT1 REG OUT;
     FJK11  (SCNT0,JSCNT0,KSCNT0,ICLK);
     FJK11  (SCNT1,JSCNT1,KSCNT1,ICLK);
    EQUATIONS
        JSCNT0 = !SCNT0 & ST1 & !ST0
               #  !SCNT0 & !ST1 & ST0;   /////    STATE COUNTER BIT0 SET INPUT /////
        KSCNT0 = SCNT0 & ST1 & !ST0
               #  SCNT0 & !ST1 & ST0
               #  ST1 & !ST0 & SCNT1 & SCNT0
               #   !ST1 & ST0 & SCNT1 & SCNT0;   /////STATE COUNTER BIT0 RESET INPUT   /////
        JSCNT1 = !SCNT1 & SCNT0 & ST1 & !ST0
               #  !SCNT1 & SCNT0 & !ST1 & ST0; ///// STATE COUNTER BIT1 SET INPUT    ////
        KSCNT1 = SCNT1 & SCNT0 & ST1 & !ST0
               #  SCNT1 & SCNT0 & !ST1 & ST0
               #  ST1 & !ST0 & SCNT1 & SCNT0
               #   !ST1 & ST0 & SCNT1 & SCNT0;   ///// STATE COUNTER BIT0 RESET INPUT /////
    END
END;

SYM GLB  C5  1  ;
    /////       CONTROL SIGNALS GLB0        /////
    SIGTYPE RFC REG OUT;
    SIGTYPE ACC_REF REG OUT;
     FJK11  (RFC,JRFC,KRFC,ICLK);
      FJK11  (ACC_REF,JACC_REF,KACC_REF,ICLK);
    EQUATIONS
        JRFC = !ST1 & ST0 & SCNT1 & !SCNT0;   /////   REFRESH COMPLETE SET INPUT    /////
         KRFC = !ST1 & ST0 & SCNT1 & SCNT0; /////   REFRESH COMPLETE RESET INPUT   /////
        JACC_REF = !ST1 & ST0 & SCNT1 & SCNT0
                 #   IRESET;                    /////    ACCESS/REFRESH SET INPUT    /////
         KACC_REF = !ST1 & !ST0 & REFRESH & !IRESET;/////ACCESS/REFRESH RESET INPUT /////
    END
END;

SYM GLB  C1  1  ;
/////      ROW ADDRESS STROBE (RAS3,RAS2) GLB        /////
    SIGTYPE IRAS3 REG OUT;
    SIGTYPE IRAS2 REG OUT;
    EQUATIONS
        IRAS3 = !ST0 & !IA20 & IRAS3 & !IRESET       /////    REDUCED RAS3      /////
              # !ST1 & !IA21 & IRAS3 & !IRESET
              # !ST0 &  ST1 & SCNT0 & SCNT1 & IA20 & IA21 & !IRESET
              #  ST0 & !ST1 & SCNT0 & SCNT1 & !IRESET
              # !ST0 & !ST1 & IRAS3 & !IRESET
              #  ST0 & ST1 & IRAS3 & !IRESET
              #  SCNT1 & IRAS3 & !IRESET
              #  SCNT0 & IRAS3 & !IRESET;
        IRAS3.CLK = ICLK;
```

```
        IRAS2 = !ST0 & IA20 & IRAS2 & !IRESET        /////    REDUCED RAS2     /////
              # !ST1 & !IA21 & IRAS2 & !IRESET
              # !ST0 &  ST1 & SCNT0 & SCNT1 & !IA20 & IA21 & !IRESET
              #  ST0 & !ST1 & SCNT0 & SCNT1 & !IRESET
              # !ST0 & !ST1 & IRAS2 & !IRESET
              #  ST0 & ST1 & IRAS2 & !IRESET
              #  SCNT1 & IRAS2 & !IRESET
              #  SCNT0 & IRAS2 & !IRESET;
        IRAS2.CLK = ICLK;

    END
END;
SYM GLB  B7  1  ;
    /////     COLUMN ADDRESS STROBE (CAS0,CAS1) GLB0        /////
    SIGTYPE ICAS0 REG OUT;
    SIGTYPE ICAS1 REG OUT;
     FJK11  (ICAS0,JCAS0,KCAS0,ICLK);
     FJK11  (ICAS1,JCAS1,KCAS1,ICLK);
    EQUATIONS
       /////    CAS0 SET INPUT    /////
       JCAS0 = ST1 & !ST0 & !IA1 & !IA0 & SCNT1 & SCNT0
            #    IRESET;
       /////CAS0 RESET                    INPUT /////
       KCAS0 = ST1 & !ST0 & !IA1 & !IA0 & !SCNT1 & SCNT0 & !IRESET;
       /////    CAS1 SET INPUT    /////
       JCAS1 = ST1 & !ST0 & !IA1 & IA0 & SCNT1 & SCNT0
          #    IRESET;
        /////CAS1 RESET INPUT /////
        KCAS1 = ST1 & !ST0 & !IA1 & IA0 & !SCNT1 & SCNT0 & !IRESET;
    END
END;

SYM GLB  B6  1  ;
    /////     COLUMN ADDRESS STROBE (CAS2,CAS3) GLB1        /////
    SIGTYPE ICAS2 REG OUT;
    SIGTYPE ICAS3 REG OUT;
     FJK11  (ICAS2,JCAS2,KCAS2,ICLK);
     FJK11  (ICAS3,JCAS3,KCAS3,ICLK);
    EQUATIONS
      JCAS2 = ST1 & !ST0 & IA1 & !IA0 & !SCNT1 & SCNT0  /////    CAS2 SET INPUT    /////
         #    IRESET;
       ///// CAS2 RESET INPUT                 /////
       KCAS2 = ST1 & !ST0 & IA1 & !IA0 & SCNT1 & SCNT0 & !IRESET;
       JCAS3 = ST1 & !ST0 & IA1 & IA0 & !SCNT1 & SCNT0   ///// CAS3 SET INPUT   /////
          #    IRESET;
       ///// CAS3 RESET INPUT                 /////
       KCAS3 = ST1 & !ST0 & IA1 & IA0 & SCNT1 & SCNT0 & !IRESET;
    END
END;

SYM GLB  B5  1  ;
/////       CONTROL SIGNALS (ACCESS,WRITE) GLB1        /////
    SIGTYPE ACCESS REG OUT;
    SIGTYPE IWREG REG OUT;
     FJK11  (ACCESS,JACCESS,KACCESS,ICLK);
     FJK11  (IWREG,JWREG,KWREG,ICLK);
    EQUATIONS
       JACCESS = !IALE & IMIO_;       /////      MEMORY ACCESS REQUEST SET INPUT    /////
        KACCESS = ST1 & !ST0 & SCNT1 & SCNT0;/////MEMORY ACCESS REQUEST RESET INPUT/////
```

```
        JWREG = !ACCESS & IRW_                    /////    WRITE REGISTER SET INPUT    /////
              #  ST1 & !ST0 & SCNT1 & SCNT0
              #  IRESET;
        KWREG = !ACCESS & !IRW_ & !IRESET; /////     WRITE REGISTER RESET INPUT    /////
   END
END;

SYM GLB  B4  1  ;
    /////      CONTROL SIGNALS (ROW/COL,RDY)GLB2        /////
    SIGTYPE ROW_COL REG OUT;
    SIGTYPE IRDY REG OUT;
     FJK11  (ROW_COL,JROW_COL,KROW_COL,ICLK);
     FJK11  (IRDY,JRDY,KRDY,ICLK);
    EQUATIONS
    JROW_COL = ST1 & !ST0 & SCNT1 & SCNT0   /////    ROW/COL SELECT SET INPUT    /////
           #  IRESET;
    KROW_COL = ST1 & !ST0 & !SCNT1 & SCNT0 & !IRESET/////ROW/COL SELECT RESET SET
               INPUT/////
        JRDY = ST1 & !ST0 & SCNT1 & !SCNT0; /////    READY SET INPUT    /////
        KRDY = ST1 & !ST0 & SCNT1 & SCNT0;  /////    READY RESET INPUT    /////
   END
END;

SYM IOC  IO16  1  ;
    //   ADDR 12 I/O CELL W/REG. INPUT //
    XPIN  IO  XA12;
     ID1  (IA12,XA12,IICLK);
END;

SYM IOC  IO15  1  ;
    //  ADDR 11 I/O CELL W/REG. INPUT //
    XPIN  IO  XA11;
     ID1  (IA11,XA11,IICLK);
END;

SYM IOC  IO14  1  ;
    //  ADDR 10 I/O CELL W/REG. INPUT //
    XPIN  IO  XA10;
     ID1  (IA10,XA10,IICLK);
END;

SYM IOC  IO13  1  ;
    //  ADDR 9 I/O CELL W/REG. INPUT //
    XPIN  IO  XA9;
     ID1  (IA9,XA9,IICLK);
END;

SYM IOC  IO12  1  ;
    //  ADDR 8 I/O CELL W/REG. INPUT //
    XPIN  IO  XA8;
     ID1  (IA8,XA8,IICLK);
END;

SYM IOC  IO11  1  ;
    //  ADDR 7 I/O CELL W/REG. INPUT //
    XPIN  IO  XA7;
     ID1  (IA7,XA7,IICLK);
END;
SYM IOC  IO10  1  ;
    //  ADDR 6 I/O CELL W/REG. INPUT //
```

```
    XPIN   IO   XA6;
     ID1  (IA6,XA6,IICLK);
END;

SYM IOC   IO9   1   ;
    //   ADDR 5 I/O CELL W/REG. INPUT //
    XPIN   IO   XA5;
     ID1  (IA5,XA5,IICLK);
END;

SYM IOC   IO8   1   ;

    //   ADDR 4 I/O CELL W/REG. INPUT //
    XPIN   IO   XA4;
     ID1  (IA4,XA4,IICLK);
END;

SYM IOC   IO7   1   ;
    //   ADDR 3 I/O CELL W/REG. INPUT //
    XPIN   IO   XA3;
     ID1  (IA3,XA3,IICLK);

END;
SYM IOC   Y2   1   ;
    //   INPUT REGISTER CLOCK (ALE)  //
    XPIN   CLK   XICLK;
     IB1  (IICLK,XICLK);
END;

SYM IOC   IO6   1   ;
    //   ADDR 2 I/O CELL W/REG. INPUT //
    XPIN   IO   XA2;
     ID1  (IA2,XA2,IICLK);
END;

SYM IOC   IO5   1   ;
   //   ADDR 1 I/O CELL W/REG. INPUT //
   XPIN   IO   XA1;
    ID1  (IA1,XA1,IICLK);
END;

SYM IOC   IO4   1   ;
    //   ADDR 0 I/O CELL W/REG. INPUT //
    XPIN   IO   XA0;
     ID1  (IA0,XA0,IICLK);
END;

SYM IOC   IO3   1   ;
    //   READY I/O CELL, OUTPUT //
    XPIN   IO   XRDY;
     OB1  (XRDY,IRDY);
END;

SYM IOC   IO2   1   ;
    //   ADDRESS LATCH ENABLE I/O CELL //
    XPIN   IO   XALE;
     IB1  (IALE,XALE);
END;

SYM IOC   IO1   1   ;
```

```
    //  MEMORY OR I/O ACCESS //
    XPIN   IO   XMIO_;
     IB1  (IMIO_,XMIO_);
END;

SYM IOC   IO0   1   ;
    //  READ WRITE SELECTION //
    XPIN   IO   XRW_;
     IB1  (IRW_,XRW_);
END;

SYM IOC   Y0   1   ;
    //  SYSTEM CLOCK INPUT //
    XPIN   CLK   XSYS_CLK LOCK 20;
     IB1  (ICLK,XSYS_CLK);
END;

SYM IOC   IO17   1   ;
    //  ADDR 13 I/O CELL W/REG. INPUT //
    XPIN   IO   XA13;
     ID1  (IA13,XA13,IICLK);
END;

SYM IOC   IO18   1   ;
    //  ADDR 14 I/O CELL W/REG. INPUT //
    XPIN   IO   XA14;
     ID1  (IA14,XA14,IICLK);
END;

SYM IOC   IO19   1   ;
    //  ADDR 15 I/O CELL W/REG. INPUT //
    XPIN   IO   XA15;
     ID1  (IA15,XA15,IICLK);
END;

SYM IOC   IO20   1   ;

    //  ADDR 16 I/O CELLõ•!
```

pLSI & ispLSI: A Multiple Function Solution

January 1992 **Application Note 2503**

Introduction

As high density programmable logic becomes more common place, determining exactly which functions to integrate and how to integrate these functions becomes more challenging. Some of the obvious considerations when integrating a design include speed and density. Beyond these concerns several other design and system details must be evaluated. In the following example, these design details will be examined and fully addressed. Design considerations can be broken into the following hierarchy: 1) System considerations including technology, reliability, and testability. 2) Design considerations which include partitioning a design for a specific architecture, determining I/O, and speed concerns. 3) Integration of the design into an ispLSI device. This includes utilizing the pLSI and ispLSI architecture for the best speed and efficient random logic consolidation.

A Dual Processor Controller

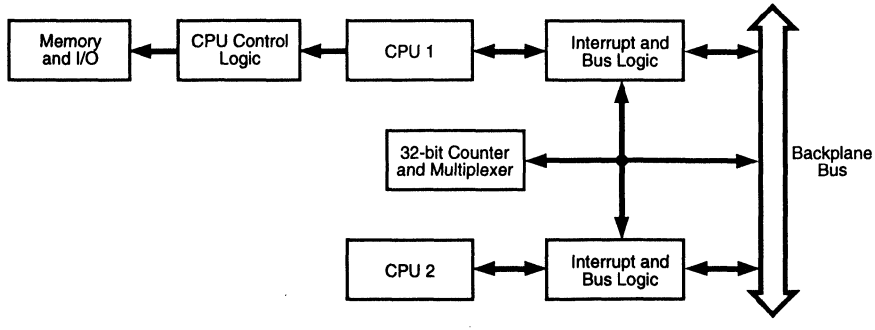
The design shown in figure 8-33 is a dual processor controller which sits on a backplane bus to which other CPUs have access. All of the CPUs communicate via the backplane bus by sending interrupts back and forth. This design also contains an independent 32-bit general purpose counter along with CPU control logic for memory and I/O.

Before partitioning the design, one must consider board space and reliability. For example, in some systems where board space and reliability are at a premium, it may be desirable to surface mount all components. In these cases, using sockets may be necessary to minimize manufacturing problems for the programmable devices. Of course, all Lattice ispLSI devices are in-system programmable, so removing devices from the board is not necessary if reprogramming is required. Another benefit to directly soldering components on the board is that less board space is needed and less capacitance will load the outputs. Therefore, soldering devices directly on the board will not increase propogation delay. To reprogram the ispLSI device, 5 volts and a five wire interface are all that is needed. In addition, choosing an instantly reprogrammable technology allows complete testability. Lattice tests for and guarantees 100% AC, DC, functional and programming yields.

Having considered these overall issues, we can now look at partitioning the design. Many designers partition by using GAL devices or other PLDs for speed and fast state machine control, and FPGAs for interface and random logic. The Lattice ispLSI family rewrites these basic design rules. With the Lattice pLSI and ispLSI families of High Density Programmable Devices, the designer acquires speed and density in one device! The design must still be partitioned, but within the Generic Logic Blocks of the Lattice ispLSI device instead of between several discrete PLDs and/or FPGAs.

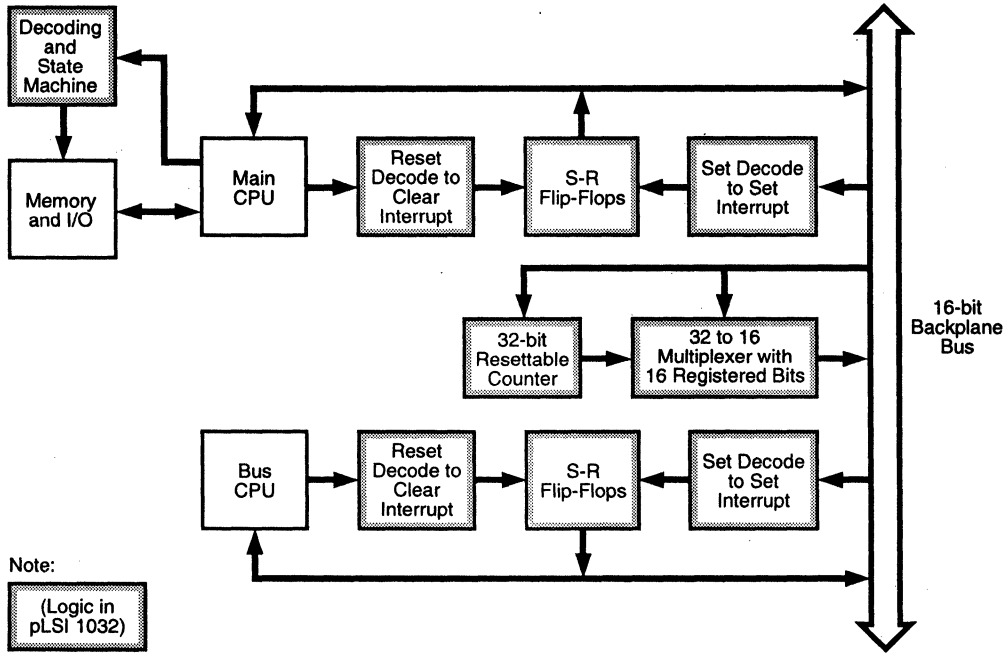
This design can be broken up into three major blocks: the two interrupt and bus random logic blocks, the data block consisting of a 32-bit counter with a 32-to-16 multiplexer, and the memory and I/O control logic state machine.

Figure 8-33. Dual Process Controller Block Diagram



LATTICE SEMICONDUCTOR CORP., 5555 Northeast Moore Ct., Hillsboro, Oregon 97124, U.S.A.
Tel. 1-800-LATTICE (528-8423); FAX (503) 681-3037

Figure 8-34. The Partitioned Design



Traditional FPGA devices would integrate the interrupt and bus logic and the 32-bit counter since the speed is not critical. The memory and I/O control logic would be left to the GAL devices. With the Lattice pLSI architecture's density and speed, many designs of this type can be fully integrated into one device.

Because of the architecture of the pLSI and ispLSI devices, the key concern for engineers will be I/O pin conservation. Counting the I/Os in this design (62 including the clocks), the pLSI 1032 with 64 I/O pins and 8 dedicated inputs will fit this application nicely. There are 4 types of input/output configurations which can be implemented by the pLSI 1032 architecture. These configurations are input only, output only, 3-state output, and bi-directional I/O. In addition, input registers and latches are also available. When executing designs with the Lattice software, it is necessary to label all of the I/O signals. I/O examples will follow later in this article. All equations are in a syntax format which can be used in an ASCII text file and then imported into, or used directly in the Lattice pLSI/ispLSI Development Systems (pDS) environment. Figure 8-34 shows the portion of design implemented in pLSI 1032 device.

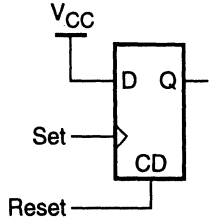
Interrupt and Bus Random Logic

Let us examine the details of each of the three sections to see how they would be implemented into the pLSI architecture. First, how to implement the decoding and latching of the interrupts. For this design, integration of the decoding logic for the set and reset terms and set/reset flip-flops is necessary.

The decoding logic is easily integrated, because the architecture has the familiar AND-OR structure. The less obvious detail which must be dealt with is exactly how to perform the Set/Reset flip-flop function. There are two choices to be explored. The first would be to use the product term reset in the Generic Logic Block, or GLB, as reset and use the product term clock as the preset with the "D" input tied to a "1." (see figure 8-35).

This approach works fine to implement a small number of unique S-R flip-flops. If many unique S-R flip-flops are needed (this example requires 12), a different implementation must be used. This is because each Generic Logic Block has all four registers sharing a common clock. Therefore using just one register would require the other

Figure 8-35. D-Type flip-flop Configured as an S-R flip-flop



three outputs to be combinatorial, or registers with the same clock and reset. If there are several unique S-R flip-flops, each would have to exist in separate GLBs. This is not an efficient use of the architecture, unless the other outputs can be used as combinatorial logic. For this example, a more effective use of the GLBs can be achieved by making an S-R flip-flop from gates. The logic equations necessary are shown in Listing 5.

With this implementation, two S-R registers can fit into one GLB. The limiting factor in deciding whether two registers will fit, is the number of inputs necessary to perform the S-R function. Each GLB has a maximum of 18 inputs. If the number of inputs (including fast feedbacks), for the two registers is 18 or less, then both equations can be used in one GLB. In this design we have a total of 12 S-R registers. Listing 5 shows the equations for two S-R registers from the design, followed by the same equations reconfigured using the gate S-R implementation in Listing 6.

The number of unique input and feedback signals in the 4 equations above, is 14. Since this is less than 18, the equations will fit in one GLB. To implement the other 10 S-R registers, simply use the same strategy and partition the logic into five other GLBs.

Data Path: 32-bit Counter and 32-to-16 Multiplexer

The next task is deciding how to build the 32-bit counter and the 32-to-16 multiplexer data latch. Using the ispLSI architecture, counter implementations up to 16-bits are straightforward. Up to 16-bits, the counter can run at the full speed of the device. Two reasons the counter is able to execute at full speed are: 1) the wide input GLBs, and 2) T-type flip-flops configurable in the architecture. The T-type flip- flop is created by inserting an XOR gate before a D-type flip-flop and feeding back the D output into one of the two inputs to the XOR gate. The other input to the XOR gate becomes the T-type flip-flop input. Beyond 16-bits, a counter must be cascaded into another level of logic because the total number of inputs needed exceeds the maximum allowed by the GLB architecture. Recall that each GLB has an 18 input limit. Two of the inputs are dedicated input pins and the other 16 are I/O pins or fast feedbacks. Therefore, to implement a 32-bit counter, we must use two more GLBs to decode the point at which the counter has reached the full 16-bit mark. This is accomplished by setting an output true when all bits (0 - 15) are a "1." Also, it is necessary to decode the point at which the counter has reached the full 24-bit mark. This is done by setting an output true when all bits (0-23) are a "1." Using these intermediate terminal count outputs, a 32-bit counter can be implemented in 10 GLBs. This 32-bit counter can run at 40 MHz as implemented here, or up to 80 MHz if the carry out is pipelined. The equations for this counter are shown in Listing 7.

8

Listing 5.

```
Q = !Set # !Qbar;          // Q is the output of the S-R  flip-flop
Qbar = !Reset # !Q;     // Qbar is the inversion of Q
```

Listing 6.

```
reset1 = bp_int_clr & bp_data12 # bp_reset;
reset2 = bp_int_clr & bp_data11 # bp_reset;
set1 = !m_as & !ipc_int & mdata8 & !mdata10 & !mdata11 & !mdata12;
set2 = !m_as & !ipc_int & mdata8 & mdata10 & !mdata11 & !mdata12;
These equations are now optimized to combine the logic in one GLB:
Q1 = !(!m_as & !ipc_int & mdata8 & !mdata10 & !mdata11 & !mdata12) # !Q1bar;   // !set1
Q1bar = !(bp_int_clr & bp_data12 # bp_reset) # !Q1 ;   //!reset1
Q2 = !(!m_as & !ipc_int & mdata8 & mdata10 & !mdata11 & !mdata12) #!Q2bar;   // !set2
Q2bar = !(bp_int_clr & bp_data11 # bp_reset) # !Q2;   // !reset2
```

Listing 7.

```
// 0-7 decode
TC_1 = (QQ_0 & QQ_1 & QQ_2 & QQ_3 & QQ_4 & QQ_5 & QQ_6 & QQ_7);
// 0-15 decode
TC_2 = (QQ_8 & QQ_9 & QQ_10 & QQ_11 & QQ_12 & QQ_13 & QQ_14 & QQ_15 & TC_1);
// 0-23 decode
TC_3 = (QQ_16&QQ_17&QQ_18 & QQ_19 & QQ_20 & QQ_21 & QQ_22 & QQ_23 & TC_2);
// The QQ_0 to QQ_31 signals are the 32 counter output bits.
QQ_0 = QQ_0 $$ VCC ;
QQ_1 = QQ_1 $$ QQ_0;
QQ_2 = QQ_2 $$ QQ_1 & QQ_0 ;
QQ_3 = QQ_3 $$ QQ_2 & QQ_1 & QQ_0 ;
QQ_4 = QQ_4 $$ QQ_3 & QQ_2 & QQ_1 & QQ_0 ;
QQ_5 = QQ_5 $$ QQ_4 & QQ_3 & QQ_2 & QQ_1 & QQ_0 ;
QQ_6 = QQ_6 $$ QQ_5 & QQ_4 & QQ_3 & QQ_2 & QQ_1 & QQ_0 ;
QQ_7 = QQ_7 $$ QQ_6 & QQ_5 & QQ_4 & QQ_3 & QQ_2 & QQ_1 & QQ_0 ;
QQ_8 = QQ_8 $$ TC_1 ;
QQ_9 = QQ_9 $$ QQ_8 & TC_1 ;
QQ_10 = QQ_10 $$ QQ_9 & QQ_8 & TC_1 ;
QQ_11 = QQ_11 $$ QQ_10 & QQ_9 & QQ_8 & TC_1 ;
QQ_12 = QQ_12 $$ QQ_11 & QQ_10 & QQ_9 & QQ_8 & TC_1 ;
QQ_13 = QQ_13 $$ QQ_12 & QQ_11 & QQ_10 & QQ_9 & QQ_8 & TC_1 ;
QQ_14 = QQ_14 $$ QQ_13 & QQ_12 & QQ_11 & QQ_10 & QQ_9 & QQ_8 & TC_1 ;
QQ_15 = QQ_15 $$ QQ_14 & QQ_13 & QQ_12 & QQ_11 & QQ_10 & QQ_9 & QQ_8 & TC_1 ;
QQ_16 = QQ_16 $$ TC_2 ;
QQ_17 = QQ_17 $$ QQ_16 & TC_2 ;
QQ_18 = QQ_18 $$ QQ_17 & QQ_16 & TC_2 ;
QQ_19 = QQ_19 $$ QQ_18 & QQ_17 & QQ_16 & TC_2 ;
QQ_20 = QQ_20 $$ QQ_19 & QQ_18 & QQ_17 & QQ_16 & TC_2 ;
QQ_21= QQ_21 $$ QQ_20 & QQ_19 & QQ_18 & QQ_17 & QQ_16 & TC_2 ;
QQ_22= QQ_22 $$ QQ_21 & QQ_20 & QQ_19 & QQ_18 & QQ_17 & QQ_16 & TC_2;
QQ_23= QQ_23 $$ QQ_22 & QQ_21 & QQ_20 & QQ_19 & QQ_18 & QQ_17 & QQ_16 & TC_2;
QQ_24= QQ_24 $$ TC_3 ;
QQ_25= QQ_25 $$ QQ_24 & TC_3 ;
QQ_26= QQ_26 $$ QQ_25 & QQ_24 & TC_3 ;
QQ_27= QQ_27 $$ QQ_26 & QQ_25 & QQ_24 & TC_3 ;
QQ_28= QQ_28 $$ QQ_27 & QQ_26 & QQ_25 & QQ_24 & TC_3 ;
QQ_29= QQ_29 $$ QQ_28 & QQ_27 & QQ_26 & QQ_25 & QQ_24 & TC_3 ;
QQ_30= QQ_30 $$ QQ_29 & QQ_28 & QQ_27 & QQ_26 & QQ_25 & QQ_24 & TC_3 ;
QQ_31= QQ_31 $$ QQ_30 & QQ_29 & QQ_28 & QQ_27 & QQ_26 & QQ_25 & QQ_24 & TC_3 ;
```

The 32-to-16 multiplexer latch is the next logic block to be constructed. In this design, the multiplexer allows the system bus access to 16-bits of the counter at a time. Either the high 16-bits (16-31) or the low 16-bits (0-15) are enabled to the bus. Since this multiplexer latch is a simple OR gate control function into a register, these 16-bits can be placed into 4 GLBs. Recall that each GLB has a maximum of four outputs. The equations for one GLB are shown in listing 8.

These 16-bits are also 3-stated by a control pin. In the ispLSI 1032 architecture, 4 unique output enable terms are allowed. Each output enable can control up to 16 outputs or bi-directional pins. For example, a design could have 64 3-state outputs, but 4 output enable control signals would be used to control 16 outputs each. It is important to note that if an output enable signal is to control more than 16 outputs, the output enable signal will need to be defined more than once. In this design only 16 outputs are controlled by one output enable signal, therefore only one output enable is used. This signal is provided by defining the output enable in a GLB as shown in listing 9.

Memory and I/O State Machine

Consider the memory and I/O state machine and decoding logic. The ispLSI GLB architecture has a path which is optimal for decoding logic. This path is utilized by choosing the 4 product term bypass mode. This mode allows an output with 4 product terms or less to exhibit input pin to output pin propagation delays of no more than 15 ns! Since decoding logic typically uses 4 product terms or less, this mode can be used for the critical propagation delay paths. The designer is cautioned to use the 4 Product Term Bypass Mode sparingly, because too many paths designated as critical in any one design may result in a failure of the Place and Route algorithm. The syntax necessary to invoke the 4 product term bypass mode is shown in listing 10.

Listing 8.

```
OMDATA0I.CLK = CLK;

OMDATA0I = (!CNTELO & QQ_16 )   #   (CNTELO & QQ_0   ); //select high or low word

OMDATA1I = (!CNTELO & QQ_17 )   #   (CNTELO & QQ_1   );

OMDATA2I = (!CNTELO & QQ_18 )   # (CNTELO & QQ_2   );

OMDATA3I = (!CNTELO & QQ_19 )   # (CNTELO & QQ_3   );

                        *

                        *

                        *

OMDATA31I=(!CNTELO & QQ_31 )    # (CNTELO & QQ_15);
```

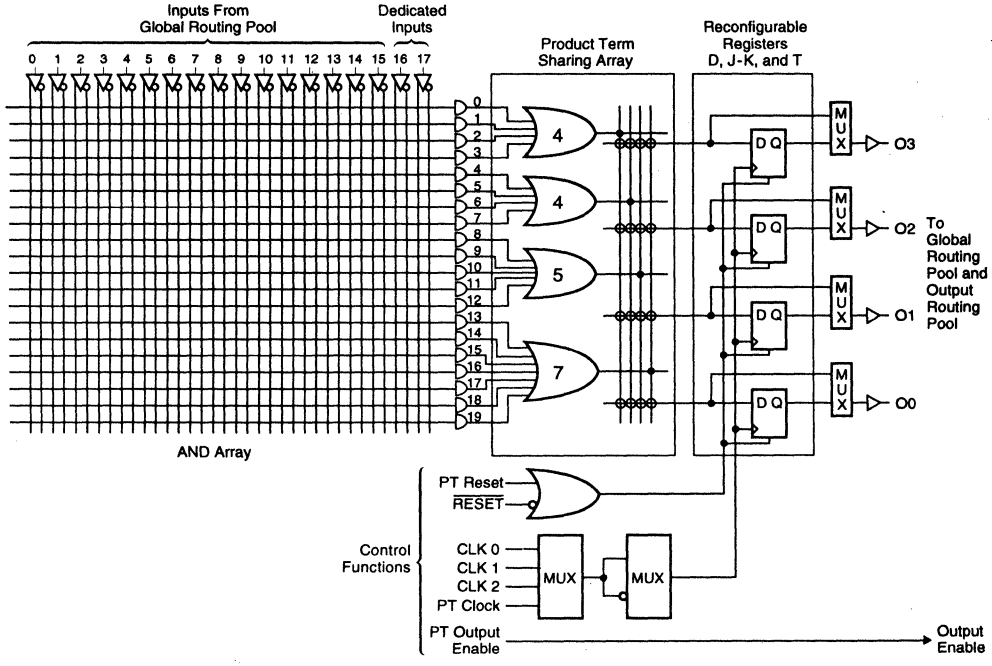
Listing 9.

```
BP_INT_RDI.OE=BP_INT_RD0;  //PROGRAMMABLE  OUTPUT  ENABLE  SIGNAL
```

Listing 10.

```
SIGTYPE   IO_SELECT 1 CRIT;   // OUT -   SIGNIFYS A COMBINATORIAL OUTPUT

//   CRIT - TELLS THE SOFTWARE TO USE THE   4   PRODUCT TERM BYPASS MODE

EQUATIONS

IO_SELECT 1= MA23 & MPA22 & !MPA21 & MPIO_ME & !MPWR_RD #
MPA23&!MPA22&MPA21&MPIO_MEM&MPWR_RD;

END;
```

Figure 8-36. GLB Product Term Sharing Array



The ispLSI 1032 is ideal for state machine applications because of two specific features. First, the I/O cell can be used to register or latch input signals. This attribute gives designers assurance that setup times to GLB registers will not be violated and metastability concerns are greatly diminished. The syntax to configure an I/O cell as a registered input is shown in listing 11.

The second feature which configures efficient state machines is the standard GLB configuration with 4, 4, 5 and 7 product terms (see figure 8-36). The product terms can be tied together to perform wider product term functions which are always needed for complex state machines. For example, in a state machine which has an output consisting of 9 product terms, the architecture will allow 4 of the product terms to be tied to 5 additional product terms, to add up to the total of 9, which is required by the state machine output. Any configuration of product term grouping is possible, including all twenty! That's right, if the design needs twenty product terms for one output, this is handled in one pass through just one GLB.

The key to successfully implementing state machines into the ispLSI 1032 is to utilize the 18 maximum inputs with up to 4 outputs, and the ability to tie the 20 product terms together. Intelligent use of these features permit the designer to streamline state machine design.

Conclusion

As can be illustrated from the above discussion, the ispLSI architecture provides designers with unparalleled flexibility, density and speed. ispLSI devices are dense and flexible enough to incorporate random logic. The architecture also contains 18-wide inputs and XOR capability in each GLB which enable counters to be effortlessly implemented. The 4 product term bypass mode allows designers to successfully realize high speed applications. Finally the ability to tie product terms together along with the input registers available at each I/O pin make this device ideal for state machine designs.

The complete Lattice Design File containing the Boolean equations for this design appears on the following pages.

Listing 11.

```
ID1 (BP_WRITE, BP_WRITEX,CLK) // This is a registered input defining the signal at the
                              // pin BP_WRITEX and after the input register BP_WRITE
```

Design LDF Listing

```
// tedfulla.ldf generated using Lattice pLSI xV1.00.22g

LDF 1.00.00 DESIGNLDF;
DESIGN cdx_design 1.00;
PART pLSI 1032-80LJ;
DECLARE
END;   //DECLARE

SYM GLB  A4  1  INTA52; // Here are 2 S-R flip-flops
SIGTYPE INTA2I   OUT;
SIGTYPE INTA3I   OUT;
SIGTYPE INTA2IBAR   OUT;
SIGTYPE INTA3IBAR   OUT;
SIGTYPE BP_INT_RDI  OE;
EQUATIONS
BP_INT_RDI  =  BP_INT_RDO;
INTA2I = !(!MAS & !IPC_INTI &
!MDATA12I & MDATA11I & MDATA10I & !MDATA8I) # !INTA2IBAR.PIN;
INTA2IBAR = !INTA2I.PIN # !(!BP_INT_CLRI & BP_DATA10I # RSETI);
INTA3I = !(!MAS & !IPC_INTI &
!MDATA12I & MDATA11I & !MDATA10I & MDATA8I) # !INTA3IBAR.PIN;
INTA3IBAR = !INTA3I.PIN #   !(!BP_INT_CLRI & BP_DATA15I # RSETI);
END;
END;

SYM GLB  A5  1  INTA52;
SIGTYPE INTA4I   OUT;
SIGTYPE INTA5I   OUT;
SIGTYPE INTA4IBAR   OUT;
SIGTYPE INTA5IBAR   OUT;
EQUATIONS
INTA4I = !(!MAS & !IPC_INTI &
!MDATA12I & MDATA11I & MDATA10I & !MDATA8I) # !INTA4IBAR.PIN;
INTA4IBAR = !INTA4I.PIN # !(!BP_INT_CLRI & BP_DATA14I # RSETI);
INTA5I = !(!MAS & !IPC_INTI  &
!MDATA12I & !MDATA11I & !MDATA10I & MDATA8I) # !INTA5IBAR.PIN;
INTA5IBAR = !INTA5I.PIN #   !(!BP_INT_CLRI & BP_DATA13I # RSETI);
END;
END;

SYM GLB  B3  1  INTAMP45;
SIGTYPE INTAMP4I   OUT;
SIGTYPE INTAMP5I   OUT;
SIGTYPE INTAMP4IBAR   OUT;
SIGTYPE INTAMP5IBAR   OUT;
EQUATIONS
INTAMP4I = !(!MAS & !IPC_INTI &
!MDATA12I & MDATA11I & !MDATA10I & !MDATA8I)   # !INTAMP4IBAR.PIN;
INTAMP4IBAR = !INTAMP4I.PIN # !(!MP_INT_CLRI & MP_DATA14I # RSETI);
INTAMP5I = !(!MAS & !IPC_INTI &
!MDATA12I & !MDATA11I & MDATA10I & MDATA8I) # !INTAMP5IBAR.PIN;
```

```
INTAMP5IBAR = !INTAMP5I.PIN #   !(!MP_INT_CLRI & MP_DATA13I # RSETI);
END;
END;

SYM GLB  B4  1  INTAMP23;
SIGTYPE  INTAMP2I   OUT;
SIGTYPE  INTAMP3I   OUT;
SIGTYPE  INTAMP2IBAR   OUT;
SIGTYPE  INTAMP3IBAR   OUT;
SIGTYPE  MP_INT_RDI  OE;
EQUATIONS
MP_INT_RDI = MP_INT_RDO;
INTAMP2I = !(!MAS & !IPC_INTI &
!MDATA12I & MDATA11I & MDATA10I & !MDATA8I) # !INTAMP2IBAR.PIN;
INTAMP2IBAR = !INTAMP2I.PIN # !(!MP_INT_CLRI & MP_DATA10I # RSETI);
INTAMP3I = !(!MAS & !IPC_INTI  &
!MDATA12I & MDATA11I & !MDATA10I & MDATA8I)  # !INTAMP3IBAR.PIN;
INTAMP3IBAR = !INTAMP3I.PIN #   !(!MP_INT_CLRI & MP_DATA15I # RSETI);
END;
END;

SYM GLB  B5  1  INTAMP01;
SIGTYPE  INTAMP0I   OUT;
SIGTYPE  INTAMP1I   OUT;
SIGTYPE  INTAMP0IBAR   OUT;
SIGTYPE  INTAMP1IBAR   OUT;
EQUATIONS
INTAMP0I = !(!MAS & !IPC_INTI &
!MDATA12I & !MDATA11I & MDATA10I & !MDATA8I) # !INTAMP0IBAR.PIN;
INTAMP0IBAR = !INTAMP0I.PIN # !(!MP_INT_CLRI & MP_DATA12I # RSETI);
INTAMP1I = !(!MAS & !IPC_INTI  &
!MDATA12I & MDATA11I & MDATA10I & MDATA8I) # !INTAMP1IBAR.PIN;
INTAMP1IBAR = !INTAMP1I.PIN #   !(!MP_INT_CLRI & MP_DATA11I # RSETI);
END;
END;

SYM GLB  B6  1  INTA01;
SIGTYPE  INTA0I   OUT;
SIGTYPE  INTA1I   OUT;
SIGTYPE  INTA0IBAR   OUT;
SIGTYPE  INTA1IBAR   OUT;
EQUATIONS
INTA0I = !(!MAS & !IPC_INTI &
!MDATA12I & !MDATA11I & MDATA10I & !MDATA8I) # !INTA0IBAR.PIN;
INTA0IBAR = !INTA0I.PIN # !(!BP_INT_CLRI & BP_DATA12I # RSETI);
INTA1I = !(!MAS & !IPC_INTI &
!MDATA12I & MDATA11I & MDATA10I & MDATA8I) # !INTA1IBAR.PIN;
INTA1IBAR = !INTA1I.PIN #   !(!BP_INT_CLRI & BP_DATA11I # RSETI);
END;
END;

SYM GLB  A7  1  BPIPLS;
SIGTYPE  BP_IPL0I  OUT;
```

```
SIGTYPE BP_IPL1I   OUT;
SIGTYPE BP_IPL2I   OUT;
EQUATIONS
BP_IPL0I = !INTA0I & !INTA2I & !INTA4I & BP_NMII & !DSP_INTI # !INTA0I &
!INTA2I & !INTA4I & OS_TICKI & BP_NMII # BP_NMII & !TMS_INTI;
BP_IPL1I = !INTA1I & !INTA3I & !INTA5I & OS_TICKI & BP_NMII & TMS_INTI #
BP_NMII & TMS_INTI & !DSP_INTI # INTA4I & BP_NMII & TMS_INTI #
INTA2I & BP_NMII & TMS_INTI # INTA0I & BP_NMII & TMS_INTI;
BP_IPL2I = !INTA0I & !INTA2I & !INTA4I & BP_NMII & TMS_INTI & DSP_INTI;
END;
END;

SYM GLB   B0   1   MPIPLS;
SIGTYPE  MP_IPL0I  OUT;
SIGTYPE  MP_IPL1I   OUT;
SIGTYPE  MP_IPL2I   OUT;
EQUATIONS
MP_IPL0I = !INTAMP0I & !INTAMP2I & !INTAMP4I & MP_NMII & !ROLL_TICKI # !INTAMP0I &
!INTAMP2I & !INTAMP4I & OS_TICKI & MP_NMII # MP_NMII & EXP_TICKI;
MP_IPL1I = !INTAMP1I & !INTAMP3I & !INTAMP5I & OS_TICKI & MP_NMII & !EXP_TICKI #
MP_NMII & !EXP_TICKI & ROLL_TICKI # INTAMP4I & MP_NMII & !EXP_TICKI #
INTAMP2I & MP_NMII & !EXP_TICKI # INTAMP0I & MP_NMII & !EXP_TICKI;
MP_IPL2I = !INTAMP0I & !INTAMP2I & !INTAMP4I & MP_NMII & !EXP_TICKI & !ROLL_TICKI;
END;
END;

SYM GLB   D0   1   Q03;
SIGTYPE  [QQ_0..QQ_3]  REG  OUT;
EQUATIONS
QQ_0.CLK  =  CLK;
QQ_1.CLK  =  CLK;
QQ_2.CLK  =  CLK;
QQ_3.CLK  =  CLK;
QQ_0  =  QQ_0  $$ VCC  ;
QQ_1  =  QQ_1   $$  QQ_0;
QQ_2  =  QQ_2  $$ QQ_1 & QQ_0  ;
QQ_3  =  QQ_3  $$ QQ_2 & QQ_1 & QQ_0  ;
END;
END;

SYM GLB   D1   1   Q47;
SIGTYPE  [QQ_4..QQ_7]  REG  OUT;
EQUATIONS
QQ_4.CLK  =  CLK;
QQ_5.CLK  =  CLK;
QQ_6.CLK  =  CLK;
QQ_7.CLK  =  CLK;
QQ_4  =  QQ_4  $$ QQ_3 & QQ_2 & QQ_1 & QQ_0  ;
QQ_5  =  QQ_5  $$ QQ_4 & QQ_3 & QQ_2 & QQ_1 & QQ_0  ;
QQ_6  =  QQ_6  $$ QQ_5 & QQ_4 & QQ_3 & QQ_2 & QQ_1 & QQ_0  ;
QQ_7  =  QQ_7  $$ QQ_6 & QQ_5 & QQ_4 & QQ_3 & QQ_2 & QQ_1 & QQ_0  ;
END;
END;
```

```
SYM GLB  D2  1  Q811;
SIGTYPE  [QQ_8..QQ_11]  REG  OUT;
EQUATIONS
QQ_8.CLK  =  CLK;
QQ_9.CLK  =  CLK;
QQ_10.CLK  =  CLK;
QQ_11.CLK  =  CLK;
QQ_8 = QQ_8 $$ TC_1 ;
QQ_9 = QQ_9 $$ QQ_8 & TC_1 ;
QQ_10 = QQ_10 $$ QQ_9 & QQ_8 & TC_1 ;
QQ_11 = QQ_11 $$ QQ_10 & QQ_9 & QQ_8 & TC_1 ;
END;
END;

SYM GLB  D3  1  Q1215;
SIGTYPE  [QQ_12..QQ_15]  REG  OUT;
EQUATIONS
QQ_12.CLK  =  CLK;
QQ_13.CLK  =  CLK;
QQ_14.CLK  =  CLK;
QQ_15.CLK  =  CLK;
QQ_12 = QQ_12 $$ QQ_11 & QQ_10 & QQ_9 & QQ_8 & TC_1 ;
QQ_13 = QQ_13 $$ QQ_12 & QQ_11 & QQ_10 & QQ_9 & QQ_8 & TC_1 ;
QQ_14 = QQ_14 $$ QQ_13 & QQ_12 & QQ_11 & QQ_10 & QQ_9 & QQ_8
                & TC_1 ;
QQ_15 = QQ_15 $$ QQ_14 & QQ_13 & QQ_12 & QQ_11 & QQ_10 & QQ_9
                & QQ_8 & TC_1 ;
END;
END;

SYM GLB  D4  1  TC1;
SIGTYPE  TC_1  OUT;
EQUATIONS
TC_1 = (QQ_0 & QQ_1 & QQ_2 & QQ_3 & QQ_4 & QQ_5 & QQ_6 & QQ_7);
END;
END;

SYM GLB  C3  1  TC2;
SIGTYPE  TC_2  OUT;
EQUATIONS
TC_2 = (QQ_8 & QQ_9 & QQ_10 & QQ_11 & QQ_12 & QQ_13 & QQ_14 & QQ_15
      & TC_1);
END;
END;

SYM GLB  D5  1  TC3;
SIGTYPE  TC_3  OUT;
EQUATIONS
TC_3 = (QQ_16 & QQ_17 & QQ_18 & QQ_19 & QQ_20 & QQ_21 & QQ_22
       & QQ_23 & TC_2);
END;
END;
```

```
SYM GLB  D6  1  TERM;
SIGTYPE TERMCNT  REG OUT;
EQUATIONS
TERMCNT.PTCLK =  (TC_1 & TC_2 & TC_3 & QQ_24 & QQ_25 & QQ_26 & QQ_27 & QQ_28 & QQ_29 &
QQ_30 & QQ_31);
TERMCNT = VCC;
END;
END;

SYM GLB  D7  1  Q1619;
SIGTYPE [QQ_16..QQ_19] REG OUT;
EQUATIONS
QQ_16.CLK = CLK;
QQ_17.CLK = CLK;
QQ_18.CLK = CLK;
QQ_19.CLK = CLK;
QQ_16 = QQ_16 $$ TC_2 ;
QQ_17 = QQ_17 $$ QQ_16 & TC_2 ;
QQ_18 = QQ_18 $$ QQ_17 & QQ_16 & TC_2 ;
QQ_19 = QQ_19 $$ QQ_18 & QQ_17 & QQ_16 & TC_2 ;
END;
END;

SYM GLB  C0  1  Q2023;
SIGTYPE [QQ_20..QQ_23] REG OUT;
EQUATIONS
QQ_20.CLK = CLK;
QQ_21.CLK = CLK;
QQ_22.CLK = CLK;
QQ_23.CLK = CLK;
QQ_20 = QQ_20 $$ QQ_19 & QQ_18 & QQ_17 & QQ_16 & TC_2 ;
QQ_21 = QQ_21 $$ QQ_20 & QQ_19 & QQ_18 & QQ_17 & QQ_16 & TC_2 ;
QQ_22 = QQ_22 $$ QQ_21 & QQ_20 & QQ_19 & QQ_18 & QQ_17 & QQ_16
                & TC_2;
QQ_23 = QQ_23 $$ QQ_22 & QQ_21 & QQ_20 & QQ_19 & QQ_18 & QQ_17
                & QQ_16 & TC_2;
END;
END;

SYM GLB  C1  1  Q2427;
SIGTYPE [QQ_24..QQ_27] REG OUT;
EQUATIONS
QQ_24.CLK = CLK;
QQ_25.CLK = CLK;
QQ_26.CLK = CLK;
QQ_27.CLK = CLK;
QQ_24 = QQ_24 $$ TC_3 ;
QQ_25 = QQ_25 $$ QQ_24 & TC_3 ;
QQ_26 = QQ_26 $$ QQ_25 & QQ_24 & TC_3 ;
QQ_27 = QQ_27 $$ QQ_26 & QQ_25 & QQ_24 & TC_3 ;
END;
END;
```

```
SYM GLB  C2  1  Q2831;
SIGTYPE [QQ_28..QQ_31] REG OUT;
EQUATIONS
QQ_28.CLK = CLK;
QQ_29.CLK = CLK;
QQ_30.CLK = CLK;
QQ_31.CLK = CLK;
QQ_28 = QQ_28 $$ QQ_27 & QQ_26 & QQ_25 & QQ_24 & TC_3 ;
QQ_29 = QQ_29 $$ QQ_28 & QQ_27 & QQ_26 & QQ_25 & QQ_24 & TC_3 ;
QQ_30 = QQ_30 $$ QQ_29 & QQ_28 & QQ_27 & QQ_26 & QQ_25 & QQ_24
                & TC_3 ;
QQ_31 = QQ_31 $$ QQ_30 & QQ_29 & QQ_28 & QQ_27 & QQ_26 & QQ_25
                & QQ_24 & TC_3  ;
END;
END;

SYM GLB  C4  1  MDATA23;
SIGTYPE    OMDATA2I REG OUT;
SIGTYPE    OMDATA3I REG OUT;
EQUATIONS
OMDATA2I.CLK  =  CNT_LTCH;
OMDATA2I = (!CNTELO & QQ_18  )
   # (CNTELO & QQ_2  );
OMDATA3I = (!CNTELO & QQ_19  )
   # (CNTELO & QQ_3  );
END;
END;

SYM GLB  B1  1  MDAT01;
SIGTYPE OMDATA0I  REG OUT;
SIGTYPE    OMDATA1I REG OUT;
EQUATIONS
XCNT_SEL1.OE  =  XCNT_SELI;
OMDATA0I.CLK  =  CNT_LTCH;
OMDATA0I = (!CNTELO & QQ_16 )
   # (CNTELO & QQ_0  );
OMDATA1I = (!CNTELO & QQ_17 )
   # (CNTELO & QQ_1  );
END;
END;

SYM GLB  C5  1  MDAT67;
SIGTYPE    OMDATA6I    REG OUT;
SIGTYPE    OMDATA7I    REG OUT;
EQUATIONS
OMDATA6I.CLK  =  CNT_LTCH;
OMDATA6I = (!CNTELO & QQ_22  )
   # (CNTELO & QQ_6  );
OMDATA7I = (!CNTELO & QQ_23  )
   # (CNTELO & QQ_7  );
END;
END;
```

```
SYM GLB   B2   1   MDAT45;
SIGTYPE OMDATA4I    REG OUT;
SIGTYPE     OMDATA5I    REG OUT;
EQUATIONS
OMDATA4I.CLK  =  CNT_LTCH;
OMDATA4I = (!CNTELO & QQ_20   )
    # (CNTELO & QQ_4   );
OMDATA5I = (!CNTELO & QQ_21   )
    # (CNTELO & QQ_5   );
END;
END;

SYM GLB   C6   1   MDAT811;
SIGTYPE OMDATA8I    REG OUT;
SIGTYPE     OMDATA9I    REG OUT;
SIGTYPE     OMDATA10I    REG OUT;
SIGTYPE     OMDATA11I    REG OUT;
EQUATIONS
XCNT_SEL.OE  =  XCNT_SELI;
OMDATA8I.CLK  =  CNT_LTCH;
OMDATA8I = (!CNTELO & QQ_24 )
    # (CNTELO & QQ_8   );
OMDATA9I = (!CNTELO & QQ_25   )
    # (CNTELO & QQ_9   );
OMDATA10I = (!CNTELO & QQ_26 )
     # (CNTELO & QQ_10   );
OMDATA11I = (!CNTELO & QQ_27 )
     # (CNTELO & QQ_11   );
END;
END;

SYM GLB   C7   1   MDAT1215;
SIGTYPE OMDATA12I    REG OUT;
SIGTYPE     OMDATA13I    REG OUT;
SIGTYPE     OMDATA14I    REG OUT;
SIGTYPE     OMDATA15I    REG OUT;
EQUATIONS
OMDATA12I.CLK  =  CNT_LTCH;
OMDATA12I = (!CNTELO & QQ_28   )
     # (CNTELO & QQ_12   );
OMDATA13I = (!CNTELO & QQ_29   )
     # (CNTELO & QQ_13   );
OMDATA14I = (!CNTELO & QQ_30 )
     # (CNTELO & QQ_14   );
OMDATA15I = (!CNTELO & QQ_31)
     # (CNTELO & QQ_15   );
END;
END;

SYM GLB   A1   1   IOMEMOE;
SIGTYPE IO_SELECT0    OUT;
SIGTYPE     IO_SELECT1    OUT CRIT;
SIGTYPE     MEMOE       OUT;
```

```
EQUATIONS
IO_SELECT0 = MPA23 & MPA22 & MPA21 &
             MPIO_MEM & MPWR_RD;
IO_SELECT1 = MPA23 & MPA22 & !MPA21 &
             MPIO_MEM & MPWR_RD;
MEMOE = !MPIO_MEM & !MPWR_RD & MPRDY;
END;
END;

SYM GLB  A0  1  MEMCSWR;
SIGTYPE MEMCS  REG OUT;
SIGTYPE   MEMWR  REG OUT;
EQUATIONS
MEMCS.CLK = CLK;
MEMCS = MPA23 & !MPA22 & !MPA21 & !MPIO_MEM #
    MEMCS & MPRDY;
MEMWR = MPA23 & !MPA22 & !MPA21 &
    !MPIO_MEM & MPWR_RD # MEMWR &
    MPRDY;
END;
END;

SYM IOC  IO51  1  MPIPLS;
XPIN IO DSP_INT;
IB1  (DSP_INTI,DSP_INT);
END;

SYM IOC  IO0  1  MPDAT15;
XPIN IO MP_DATA15          ;
BI1  (MP_DATA15I,MP_DATA15,INTAMP5I,MP_INT_RDI);
END;

SYM IOC  IO1  1  MPDAT14;
XPIN IO MP_DATA14          ;
BI1  (MP_DATA14I,MP_DATA14,INTAMP4I,MP_INT_RDI);
END;

SYM IOC  IO2  1  MPDAT13;
XPIN IO MP_DATA13          ;
BI1  (MP_DATA13I,MP_DATA13,INTAMP3I,MP_INT_RDI);
END;

SYM IOC  IO3  1  MPDAT12;
XPIN IO MP_DATA12          ;
BI1  (MP_DATA12I,MP_DATA12,INTAMP2I,MP_INT_RDI);
END;

SYM IOC  IO4  1  MPDAT11;
XPIN IO MP_DATA11          ;
BI1  (MP_DATA11I,MP_DATA11,INTAMP1I,MP_INT_RDI);
END;
```

```
SYM IOC  IO5  1  MPDAT10;
XPIN IO MP_DATA10            ;
BI1  (MP_DATA10I,MP_DATA10,INTAMP0I,MP_INT_RDI);
END;

SYM IOC  IO8  1  MPINTCL;
XPIN IO MP_INT_CLR    LOCK 40 ;
IB1  (MP_INT_CLRI,MP_INT_CLR);
END;

SYM IOC  IO9  1  RSET;
XPIN IO RSET   ;
IB1  (RSETI,RSET);
END;

SYM IOC  IO10  1  MDATA15;
XPIN IO MDATA15     LOCK 53 ;
OT1  (MDATA15,OMDATA15I,!XCNT_SEL);
END;

SYM IOC  IO11  1  MDATA14;
XPIN IO MDATA14     LOCK 54 ;
OT1  (MDATA14,OMDATA14I,!XCNT_SEL);
END;

SYM IOC  IO12  1  MDATA13;
XPIN IO MDATA13     LOCK 55 ;
OT1  (MDATA13,OMDATA13I,!XCNT_SEL);
END;

SYM IOC  IO13  1  MDATA12;
XPIN IO MDATA12     LOCK 56 ;
BI1  (MDATA12I,MDATA12,OMDATA12I,!XCNT_SEL);
END;

SYM IOC  IO14  1  MDATA11;
XPIN IO MDATA11     LOCK 57 ;
BI1  (MDATA11I,MDATA11,OMDATA11I,!XCNT_SEL);
END;

SYM IOC  IO15  1  MDATA10;
XPIN IO MDATA10     LOCK 58 ;
BI1  (MDATA10I,MDATA10,OMDATA10I,!XCNT_SEL);
END;

SYM IOC  IO16  1  MDATA9;
XPIN IO MDATA9     LOCK 59 ;
OT1  (MDATA9,OMDATA9I,!XCNT_SEL);
END;

SYM IOC  IO17  1  MDATA8;
XPIN IO MDATA8     LOCK 60 ;
BI1  (MDATA8I,MDATA8,OMDATA8I,!XCNT_SEL);
END;
```

```
SYM IOC  IO26  1  MAS;
XPIN IO MASX          LOCK 27 ;
IB1  (MAS,MASX);
END;

SYM IOC  IO27  1  IPC_INT;
XPIN IO IPC_INT      LOCK 26 ;
IB1  (IPC_INTI,IPC_INT);
END;

SYM IOC  IO28  1  MPIPL2;
XPIN IO MP_IPL2;
OB1  (MP_IPL2,MP_IPL2I);
END;

SYM IOC  IO29  1  MPIPL1;
XPIN IO MP_IPL1;
OB1  (MP_IPL1,MP_IPL1I);
END;

SYM IOC  IO30  1  MPIPL0;
XPIN IO MP_IPL0;
OB1  (MP_IPL0,MP_IPL0I);
END;

SYM IOC  IO31  1  MPINTRD;
XPIN IO MP_INT_RD;
IB1  (MP_INT_RDO,MP_INT_RD);
END;

SYM IOC  IO32  1  BPINTRD;
XPIN IO BP_INT_RD;
IB1  (BP_INT_RDO,BP_INT_RD);
END;

SYM IOC  IO33  1  BPDAT15;
XPIN IO BP_DATA15    ;
BI1  (BP_DATA15I,BP_DATA15,INTA5I,BP_INT_RDI);
END;

SYM IOC  IO34  1  BPDAT14;
XPIN IO BP_DATA14    ;
BI1  (BP_DATA14I,BP_DATA14,INTA4I,BP_INT_RDI);
END;

SYM IOC  IO35  1  BPDAT13;
XPIN IO BP_DATA13    ;
BI1  (BP_DATA13I,BP_DATA13,INTA3I,BP_INT_RDI);
END;

SYM IOC  IO36  1  BPDAT12;
XPIN IO BP_DATA12    ;
BI1  (BP_DATA12I,BP_DATA12,INTA2I,BP_INT_RDI);
END;
```

```
SYM IOC  IO37  1  BPDAT11;
XPIN IO BP_DATA11   ;
BI1  (BP_DATA11I,BP_DATA11,INTA1I,BP_INT_RDI);
END;

SYM IOC  IO38  1  BPDAT10;
XPIN IO BP_DATA10   ;
BI1  (BP_DATA10I,BP_DATA10,INTA0I,BP_INT_RDI);
END;

SYM IOC  IO41  1  BPINTCL;
XPIN IO BP_INT_CLR  ;
IB1  (BP_INT_CLRI,BP_INT_CLR);
END;

SYM IOC  IO42  1  BPIPL2;
XPIN IO BP_IPL2 ;
OB1  (BP_IPL2,BP_IPL2I);
END;

SYM IOC  IO43  1  BPIPL1;
XPIN IO BP_IPL1 ;
OB1  (BP_IPL1,BP_IPL1I);
END;

SYM IOC  IO44  1  BPIPL0;
XPIN IO BP_IPL0 ;
OB1  (BP_IPL0,BP_IPL0I);
END;

SYM IOC  IO45  1  MP_NMI;
XPIN IO MP_NMI;
IB1  (MP_NMII,MP_NMI);
END;

SYM IOC  IO46  1  OS_TICK;
XPIN IO OS_TICK;
IB1  (OS_TICKI,OS_TICK);
END;

SYM IOC  IO47  1  EXPTICK;
XPIN IO EXP_TICK;
IB1  (EXP_TICKI,EXP_TICK);
END;

SYM IOC  IO48  1  ROLTICK;
XPIN IO ROLL_TICK;
IB1  (ROLL_TICKI,ROLL_TICK);
END;

SYM IOC  IO49  1  BP_NMI;
XPIN IO BP_NMI;
IB1  (BP_NMII,BP_NMI);
END;
```

```
SYM IOC   IO50   1   TMS_INT;
XPIN  IO  TMS_INT;
IB1   (TMS_INTI,TMS_INT);
END;

SYM IOC   IO18   1   MPCLR13;
XPIN IO MDATA7          ;
OT1   (MDATA7,OMDATA7I,!XCNT_SEL1);
END;

SYM IOC   IO19   1   MPCLR13;
XPIN IO MDATA6          ;
OT1   (MDATA6,OMDATA6I,!XCNT_SEL1);
END;

SYM IOC   IO20   1   MPCLR13;
XPIN IO MDATA5                    LOCK 6 ;
OT1   (MDATA5,OMDATA5I,!XCNT_SEL1);
END;

SYM IOC   IO21   1   MPCLR13;
XPIN IO MDATA4                    LOCK 5 ;
OT1   (MDATA4,OMDATA4I,!XCNT_SEL1);
END;

SYM IOC   IO22   1   MPCLR13;
XPIN IO MDATA3      ;
OT1   (MDATA3,OMDATA3I,!XCNT_SEL1);
END;

SYM IOC   IO23   1   MPCLR13;
XPIN IO MDATA2      ;
OT1   (MDATA2,OMDATA2I,!XCNT_SEL1);
 END;

SYM IOC   IO24   1   MPCLR13;
XPIN IO MDATA1           LOCK 4 ;
OT1   (MDATA1,OMDATA1I,!XCNT_SEL1);
END;

SYM IOC   IO25   1   MPCLR13;
XPIN IO MDATA0           LOCK 3 ;
OT1   (MDATA0,OMDATA0I,!XCNT_SEL1);
END;

SYM IOC   IO63   1   LTCH;
XPIN   IO   XCNTSEL;
IB1    (XCNT_SELI,    XCNTSEL);
END;

SYM IOC   Y1   1   LTCH;
XPIN  CLK  LTCH;
IB1   (CNT_LTCH,LTCH);
END;
```

```
SYM IOC  Y0 1  CLOCK;
XPIN  CLK  XCLK;
IB1 (CLK,  XCLK);
END;

SYM IOC  IO62  1  CNTELO;
XPIN  IO  OCNTELO  ;
IB1  (CNTELO, OCNTELO);
END;

SYM IOC  IO61  1  TERMCNT;
XPIN  IO  XTERMCNT;
OB1  (XTERMCNT,  TERMCNT);
END;

SYM IOC  IO59  1  MPA23;
XPIN IO MPA23O;
IB1  (MPA23, MPA23O);
END;

SYM IOC  IO58  1  MPA21;
XPIN IO MPA22O;
IB1  (MPA22, MPA22O);
END;

SYM IOC  IO57  1  MPA21;
XPIN IO MPA21O;
IB1  (MPA21, MPA21O);
END;

SYM IOC  IO56  1  MPIO_MEM;
XPIN IO MPIO_MEMO;
IB1  (MPIO_MEM,  MPIO_MEMO);
END;

SYM IOC  IO55  1  MPWR_RD;
XPIN IO MPWR_RDO;
ID1  (MPWR_RD,  MPWR_RDO,CLK);
END;

SYM IOC  IO54  1  MPRDY;
XPIN IO MP_RDYO;
IB1  (MPRDY,  MP_RDYO);
END;

SYM IOC  IO53  1  IO_SELECT;
XPIN IO IO_SELECT0O;
OB1  (IO_SELECT0O,  IO_SELECT0);
END;

SYM IOC  IO52  1  IO_SELECT1;
XPIN IO IO_SELECT1O;
OB1  (IO_SELECT1O,  IO_SELECT1);
END;
```

```
SYM IOC   IO7   1   MEMCS;
XPIN IO MEMCSO   ;
OB1  (MEMCSO,  MEMCS);
END;

SYM IOC   IO60   1   MEMOE;
XPIN  IO  MEMOEO;
OB1  (MEMOEO,  MEMOE);
END;
END;    //LDF  DESIGNLDF
```

Video Graphics Controller

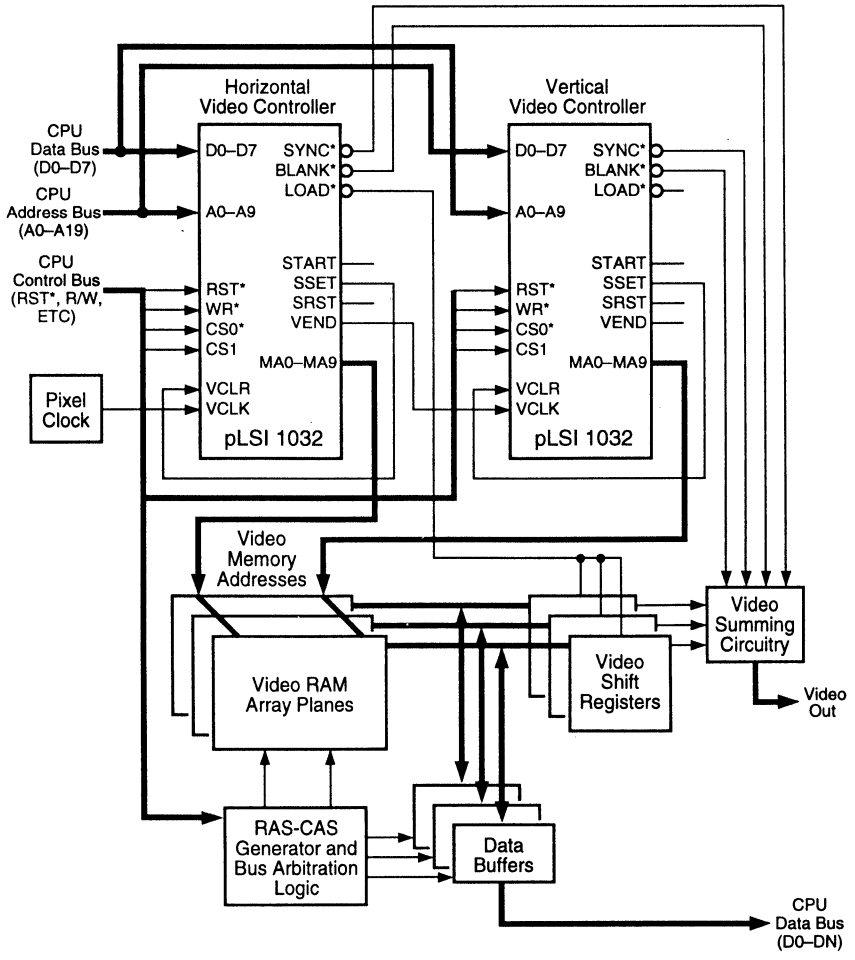
January 1992

Application Note 2504

Introduction

This Graphics Controller design consists of two pLSI 1032 chips programmed identically to produce most of the basic video functions and timing signals associated with a general purpose graphics interface. The generic design of the controller allows customization by adding additional circuitry for a Graphics Controller System based on the design specific requirements (see system block diagram, figure 8-41).

Figure 8-41. Video Graphics Controller System Block Diagram



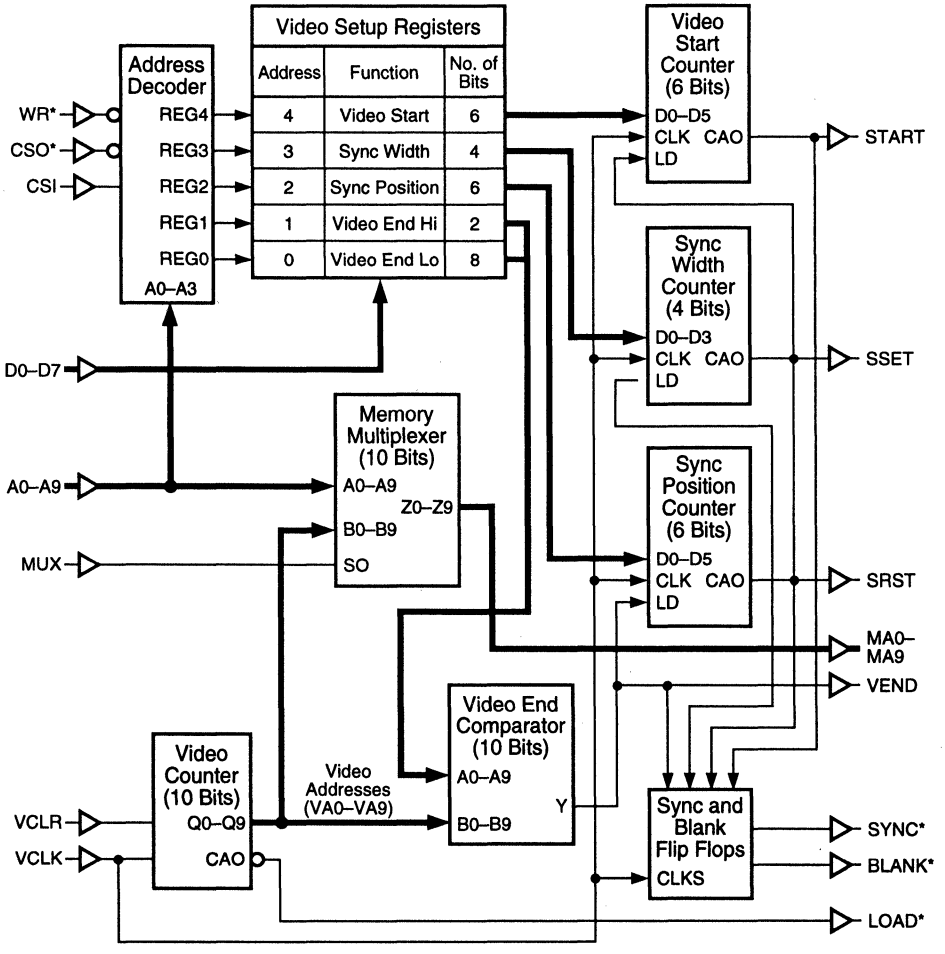
LATTICE SEMICONDUCTOR CORP., 5555 Northeast Moore Ct., Hillsboro, Oregon 97124, U.S.A.
Tel. 1-800-LATTICE (528-8423); FAX (503) 681-3037

This design is capable of a maximum 1024 X 1024 non-interlaced display with programmable blanking and sync signal positioning. One of the pLSI 1032s is used for Horizontal Video Control (HVC) and the other for Vertical Video Control (VVC). Because the two pLSI 1032s are programmed identically, the LOAD* signal (Schematic 2) is redundant on the VVC chip and only used on the HVC chip.

Referencing figure 8-42, the Video Graphics Controller Chip block diagram, the signals which the CPU sends to the Video Graphics Controller (VGC), are: WRITE (WR*), CHIP SELECT 0*/1 (CSO*/1), DATA BUS (D0-D7), ADDRESS BUS (A0-A9), and MULTIPLEXER SELECT (MUX). The Address Decoder receives an address from the CPU. Once decoded, this address enables one of the Video Setup Registers (VSRs) which then receives video information from the CPU data bus. This setup data is then fed to the appropriate counter or comparator, which actually controls that specific display parameter.

The CPU address bus is also interfaced to the Memory Multiplexer (MMUX) "A" inputs. The "B" inputs of the MMUX are connected to the outputs of the Video Counter (VCNTR). The MMUX allows either the CPU or the VCNTR to access video memory depending on the polarity of the MUX signal from the CPU. Additionally, the VCNTR produces the LOAD* signal to the video shift register, which is external to the pLSI 1032.

Figure 8-42. Video Graphics Controller Chip Block Diagram



The VCNTR also feeds the Video End Comparator (VEC). The VEC compares the addresses from the VCNTR and the Video End Hi and Lo registers which are located in the VSRs. When true, the VEC outputs the Video End (VEND) signal and simultaneously enables the load for the Sync Position Counter (SPC), while clearing the Blanking flip-flop.

The SPC data is loaded from the Sync Position register which is located in the VSRs. The SPC counts down to zero at which point it outputs the Sync Reset (SRST) signal. SRST also enables the load for the Sync Width Counter (SWC), and clears the Sync flip-flop.

The SWC's data comes from the Sync Width register in the VSRs. The SWC counts down to zero. At zero, it enables the load for the Video Start Counter (VSC), and also sets the Sync flip-flop.

The VSC receives its data from the Video Start VSR. The VSC counts down to zero, and while at zero it produces the START signal simultaneously setting the Blanking flip-flop.

1) Address Decoder (Schematic 2)

The address decoder is enabled by the WR* and CS0*/1 signals and decodes address bits A0-A2 into one of five active high select output signals, R0-R4. These are the select lines to the video attribute setup registers (schematic 3). The CS0* active low chip-select and CS1 active high chip-select are for differentiating between the horizontal controller and the vertical controller when interfacing to the CPU bus as two of these chips must be used in the system. The WR* is used to synchronize the access to the registers with CPU write cycle. All accesses to this block are write only.

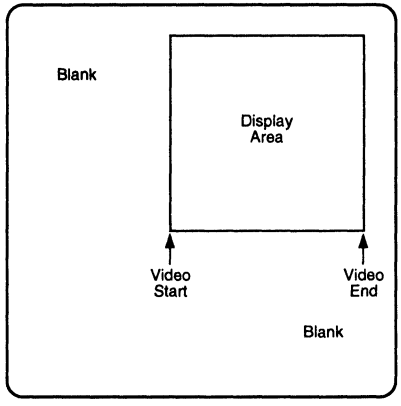
2) Video Setup Registers (Schematic 3)

The circuit is designed to interface to an 8-bit data bus but could be easily redesigned to interface to a 16-bit bus. The Video attribute Setup Register's addresses and widths are as shown in table 8-7.

Address	Name-Function	Number of bits
0	Video End Low (Ve 7:0)	8
1	Video End High (Ve 9:8)	2
2	Sync Position (Sp 5:0)	6
3	Sync Width (Sw 3:0)	4
4	Video Start (Vs 5:0)	6

These registers provide the data to be compared or loaded into one of the dead-end down counters used for positioning the display viewing area or sync pulse positions and widths (see figure 8-43).

Figure 8-43. Typical Video Display Set up



Video End Low and High Registers

These registers combine to form the 10-bit address location of the video display endpoints. In the case of the horizontal display location, this is the right hand side of the screen and the vertical display location is the bottom, or last visible scan line. In other words this is the point where video ends and blanking begins.

Sync Position Register

This 6-bit register holds the value of the distance from where video ends and the horizontal or vertical sync pulses start thus allowing for sync pulse positioning relative to video end. This is counted in pixels in the horizontal plane and lines in the vertical plane. The value of this register cannot be less than 1.

Sync Width Register

This 4-bit register holds the value of the sync pulse width. This is counted in pixels in the horizontal plane and lines in the vertical plane. The value of this register cannot be less than 1.

Video Start Register

This 6-bit register holds the value of the distance from where the sync pulse or blanking ends and video starts. This is counted in pixels in the horizontal plane and lines in the vertical plane. The value of this register cannot be less than 1.

3) Video Counter (Schematic 2)

This is a 10-bit counter which provides the video addresses VA0-VA9. In the case of the horizontal controller, this register provides the LOAD* signal for the video RAM shift registers. This register's synchronous outputs, clock, and asynchronous reset lines are accessible from the I/O pins of the chip for interfacing with the system's horizontal and vertical functions. The reset to the counter is VCLR and is typically connected externally to the SSET signal (schematic 5). SSET resets the counter at the end of the sync pulse. This can be customized for the specific application. VCLK is the clock input to the counter. VCLK is connected to the pixel clock of the horizontal controller (HVC) and is driven by the VEND signal from the horizontal controller (HVC) in the case of the vertical controller (VVC). The LOAD* signal output is a 1 cycle-wide pulse every 16 pixels. This can be reduced to 8 pixels by modifying the counter's boolean statements.

4) Video End Comparator (Schematic 5)

This is a 10-bit comparator which compares the 10-bit value in the Video End Low and High registers (R0-R1 schematic 3), to the 10-bit value of the Video Counter. When the compare is true a 1 cycle-wide pulse is generated called VEND. This is the end of visible video and starts the sync position counter running while also clearing the blanking flip-flop.

5) Sync Position Counter (Schematic 4)

This is a 6-bit loadable, dead-end down counter which counts until it reaches 0 and then holds until it is loaded with a value greater than or equal to 1. The load is activated by the VEND signal generated by the Video End Comparator. The count is a maximum of 64 pixels (horiz) or lines (vert) and is loaded each time with the value of the Sync Position Register (R2). When the count reaches zero the counter produces the signal SRST which starts the Sync Width Counter and clears the Sync flip-flop (schematic 5).

6) Sync Width Counter (Schematic 5)

This is a 4-bit loadable, dead-end down counter which counts until it reaches 0 and then holds until it is loaded with a value greater than or equal to 1. The load is activated by the SRST signal which is generated by the sync position counter. The count is a maximum of 16 pixels (horiz) or lines (vert) and is loaded each time with the value of the Sync Width Register (R3 schematic 3). When the count reaches zero the counter produces the signal SSET which starts the Video Start Counter running and sets the sync flip-flop.

7) Video Start Counter (Schematic 4)

This is a 6-bit loadable, dead-end down counter which counts until it reaches 0 and then holds until it is loaded with a value greater than or equal to 1. The load is activated by the SSET signal (schematic 5), generated by the Sync Width Counter. The count is a maximum of 64 pixels (horiz) or lines (vert) and is loaded each time with the value of the Video Start Register (R4 schematic 3). When the count reaches zero the counter produces the signal START which sets the Blanking flip-flop (schematic 5).

8) Sync flip-flop (Schematic 5)

This flip-flop is cleared by the signal SRST (schematic 4), and set by the signal SSET to produce the sync pulse for either horizontal or vertical. It is a J-K flip-flop which is clocked by VCLK that delays the actual edges by one clock. This factor must be taken into account when calculating the sync position and sync width values as the value is one less than the true position or width. These values must be no less than 1.

9) Blanking flip-flop (Schematic 5)

This flip-flop is cleared by the signal VEND and set by the signal START (schematic 4), to produce the blanking signal for either horizontal or vertical controllers. It is a J-K flip-flop which is clocked by VCLK This flip-flop delays the actual edges by 1 clock. This must be taken into account when calculating the sync position and sync width values as the value is one less than the true position or width. Thus the Sync position and width values must be greater than or equal to one.

10) Memory Address Multiplexer (Schematic 6)

This is a dual input 10-bit multiplexer which outputs either the video addresses (VA0-VA9), or the CPU addresses (A0-A9), to the output pins (MA0-MA9). This allows for either the video counters or the CPU to directly address the video memory. The multiplexer is controlled by the signal MUX and when MUX is low selects the CPU address. When MUX is high it selects the video counters (horizontal and vertical).

This system design is generic in terms of the size and number of the video memory planes. It is based on the additional support of RAS-CAS logic, if multiplexed dynamic RAM is used, along with bus arbitration logic to allow for transparent accesses by the CPU. It also assumes that the shift registers (if used), are correctly chosen and interfaced to the video RAM. The final support circuitry is video summing which, depending on the type of display to be driven (analog or digital), and the polarity of the blanking and sync signals has a wide variation of layouts. All of these functions, when finally chosen, can be easily incorporated into the additional 25% of each of the HVC and VVC chips remaining, or placed into additional pLSI devices as needed. This design allows for quick and flexible programmable video graphic interface to numerous applications.

Pin functional descriptions

NAME	TYPE	FUNCTION
WR*	Input	Allow strobe used to write data into video attribute set up register. Selected by address lines A0-A2. Also qualified with CS0*/1.
CS0*/1	Input	Active low/high chip select used to enable writes to attribute set up registers.
AO-A9	Input	A0-A2 are used to select one of the video attribute set up registers. A0-A9 are used to address the video memory.
D0-D7	Input	Data input to the video attribute set up registers.
MUX	Input	Mux select line for video memory access. High select CPU addresses (A0-A9), low select video counter addresses (VA0-VA9).
MA0-MA9	Output	Video memory address lines.
VEND	Output	Active high signal used to indicate the end of a horizontal or vertical scan.
SRST	Output	Active high signal used to indicate the end of horizontal or vertical Sync.
SSET	Output	Active high signal used to indicate the beginning of a horizontal or vertical Sync.
START	Output	Active high signal used to indicate the start of a horizontal or vertical visible scan.
LOAD*	Output	Active low signal used to load the external video shift registers with data from the video memory.
BLANK*	Output	Active low signal used to indicate the blanking of horizontal or vertical display.
SYNC*	Output	Active low signal used to indicate the horizontal or vertical Sync pulse.
VCLK	Input	System clock running at same frequency as the monitor.
VCLR	Input	Active high signal used to asynchronously reset the video counters. This allows for either horizontal or vertical operation of the device.

Video attribute formulas

The following are the formulas for calculating the display characteristics:

tc = pixel clock time period (ie: 10Mhz = 100ns)
Ve = video end (0-1024)
Sp = sync position (1-63)
Sw = sync width (1-15)
Vs = video start (1-63)

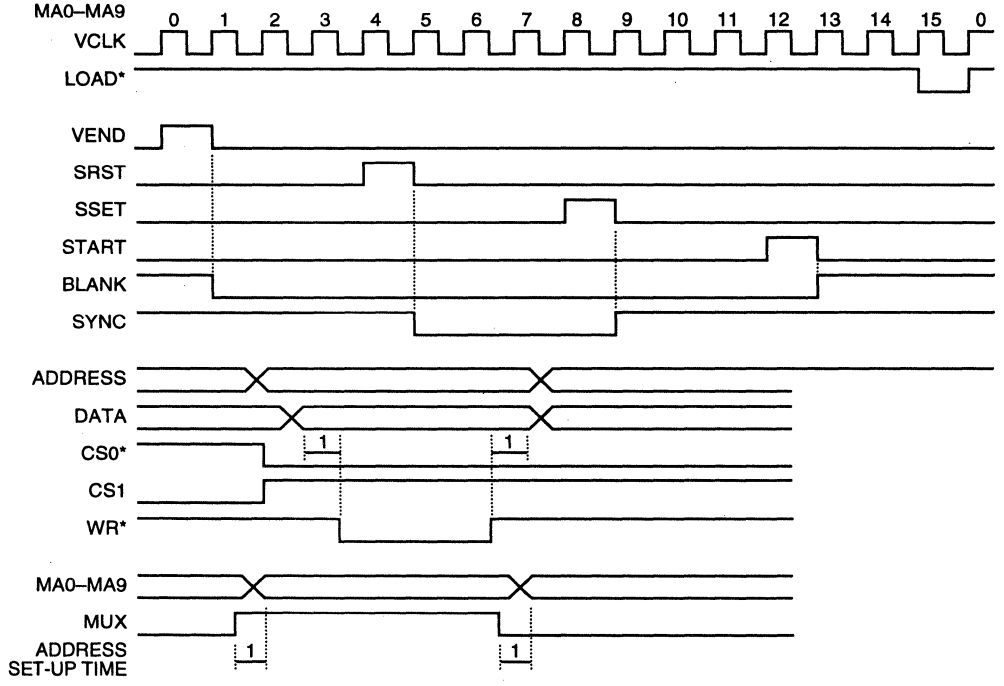
Horizontal (HVC)

- horizontal scan line period = [Ve+(Sp+1)+(Sw+1)+(Vs+1)] * tc
- horizontal scan rate = 1/horizontal scan line period
- horizontal display period = [Ve-(Vs+1)] * tc
- LOAD* frequency = tc * 1

Vertical (VVC)

- vertical scan line period = [Ve+(Sp+1)+(Sw+1)+(Vs+1)] * horizontal scan line period
- vertical scan rate = 1/vertical scan line period
- vertical display period = [Ve-(Vs+1)] *horizontal scan line period

Figure 8-44. Video Graphics Controller Timing



1. Note:
   See Sidebar for Description

The major timing relationships for this device are shown in figure 8-44. All signals are shown in relation to VCLK.

As can be seen from the diagram, LOAD* is generated every 16 VCLKs. LOAD* loads the video shift registers with data from the video memory. BLANK is activated by the falling edge of VEND and is inactivated at the falling edge of START. SYNC goes low at the falling edge of SRST and rises with the falling edge of SSET.

The CPU related signals are shown in waveforms 9 to 13. CSO* and CS1 are really complimentary versions of the same signal. Because two pLSI 1032s are used in the design, CS0*, for example, would be used as the chip select for the horizontal controller chip CS1 would then be used as the chip select for the vertical controller chip. In any case, there is a set-up and hold time associated with a data write into the chip. This is indicated by the short solid lines bounded by the dashed lines in between the DATA and CS0* waveforms. The actual set-up and hold times involved are dependent upon the frequency of VCLK, but the relationship to VCLK is clearly shown.

The last two waveforms on the diagram show the delay from MUX rising or falling and the validity of the addresses on MA0 to MA9. This delay employs the same caveat as above - the actual time depends upon the frequency of VCLK.

The pLSI Advantage

The pLSI 1032 is an excellent choice for this type of design because of its density, flexibility, and speed. The device utilization percentages for this particular design are: 75% GLB, 66% GLB output, and 61% I/O. This means that there is enough of the device left to interface to a 16-bit bus or to add glue logic which might be associated with a specific design. The I/O assignment in the pLSI 1032 is extremely flexible. I/Os can be fixed to a specific pin, or left for the router to decide the best connection. With no fixed pins, this design took 1.5 minutes to route. With all but twelve of the pins fixed, routing took just over 10 minutes.

The rest of this Applications Note consists of an appendix which contains the schematics and a hardcopy of the LDF file for this design.

Appendix

```
// graphfix.ldf generated using Lattice pDS software V1.00.19
LDF 1.00.00 DESIGNLDF;
DESIGN GRAPHICS 1.00;
PROJECTNAME MSD QUOTE #2128;
DESCRIPTION
This is one of two identical chips used for either horizontal or vertical control in
the graphics controller design. Two of these chips produce most of the basic
video functions and timing signals associated with a general purpose graphics
interface. The design is capable of up to a 1024 X 1024 non-interlaced display
with programmable blanking and sync signal positioning. One of the chips is
used for horizontal video control (HVC) and the other, vertical video control (VVC).;
PART pLSI1032-80LJ;
DECLARE
END;  //DECLARE

SYM GLB D4 1 MISC. SIGNALS 2;
// SSET signal generation, SYNC & BLANK;
// intermediate signal generation;
SIGTYPE SYNC REG OUT;
SIGTYPE BLANK REG OUT;
SIGTYPE SSET OUT;
      EQUATIONS
          SYNC.CLK=VCLK
          SSET=!SSET1&SSET0;
          SYNC.D = !(!(!SYNC.Q & SSET) & (!SYNC.Q # SRST));
          BLANK.D = !(!(!BLANK.Q & START) & (!BLANK.Q # VEND));
       END;
END;

SYM GLB C7 1 ENABLE - !WR&!CS0&CS1;
// Write enable qualification for address decoder;
SIGTYPE ENABLE OUT;
      EQUATIONS
          ENABLE = !WR & !CS0 & CS1;
      END;
END;

SYM GLB A1 1 VIDEO COUNTERS;
// Video memory address counter bits VA4-VA7;
SIGTYPE [VA4..VA7] REG OUT;
       EQUATIONS
          VA4.CLK = VCLK;
          VA4.RE = VCLR;
          VA4=(VA0 & VA1& VA2 & VA3) $$ VA4;
          VA5=(VA0 & VA1 & VA2 & VA3 & VA4) $$ VA5;
          VA6=(VA0 & VA1 & VA2 & VA3 & VA4 & VA5) $$ VA6;
          VA7=(VA0 & VA1 & VA2 & VA3 & VA4 & VA5 & VA6) $$ VA7;
       END;
END;
```

```
SYM GLB   A0   1   VIDEO COUNTERS;
// Video memory address counter bits VA0-VA3;
SIGTYPE  [VA0..VA3]  REG  OUT;
       EQUATIONS
          VA0.CLK = VCLK;
          VA0.RE = VCLR;
          VA0 = VA0 $$ VCC;
          VA1 = VA0 $$ VA1;
          VA2 = (VA0 & VA1) $$ VA2;
          VA3 = (VA0 & VA1& VA2) $$ VA3;
      END;
END;

SYM GLB   A2   1   VIDEO COUNTERS;
// Video memory address counter bits VA8,VA9;
//   and LOAD signal output generation;
SIGTYPE  VA8  REG  OUT;
SIGTYPE  VA9  REG  OUT;
SIGTYPE  LOAD  OUT;
       EQUATIONS
          VA8.CLK = VCLK;
          VA8.RE = VCLR;
          VA8=(VA0 & VA1 & VA2 & VA3 & VA4 & VA5 & VA6 & VA7) $$ VA8;
          VA9=(VA0 & VA1 & VA2 & VA3 & VA4 & VA5 & VA6 & VA7 & VA8) $$ VA9;
          LOAD=VA0 & VA1 & VA2 & VA3;
      END;
END;

SYM GLB   A3   1   ADDRESS DECODE;
// Register address decoder;
SIGTYPE  [R0..R3]  OUT;
       EQUATIONS
          R0 = ENABLE & !A0 & !A1 & !A2;
          R1 = ENABLE & A0 & !A1 & !A2;
          R2 = ENABLE & !A0 & A1 & !A2;
          R3 = ENABLE & A0 & A1 & !A2;
      END;
END;

SYM GLB   A4   1   END HI (VIDEO);
// R4 of register address decoder and video;
// data registers (video end hi);
SIGTYPE  R4  OUT;
SIGTYPE  [R1Q0..R1Q1]  OUT;
EQUATIONS
          R4 = !WR & !CS0 & CS1 & !A0 & !A1 & A2;
          [R1Q0..R1Q1] = [D4..D5] & R1;
   END;
END;

SYM GLB   A5   1   END LO 1 (VIDEO);
// Video data registers (video end lo);
```

```
SIGTYPE  [R0Q0..R0Q3]  OUT;
      EQUATIONS
           [R0Q0..R0Q3] = [D0..D3] & R0;
      END;
END;

SYM GLB   A6   1   END LO 2 (VIDEO);
// Video data registers (video end lo);
SIGTYPE  [R0Q4..R0Q7]  OUT;
      EQUATIONS
           [R0Q4..R0Q7] = [D4..D7] & R0;
      END;
END;

SYM GLB   A7   1   POSITION, SYNC 1;
// Video data registers (sync position);
SIGTYPE  [R2Q0..R2Q3]  OUT;
      EQUATIONS
           [R2Q0..R2Q3] = [D0..D3] & R2;
      END;
END;

SYM GLB   B0   1   START & POSITION 2;
// Video data registers (sync position);
// Video data registers (video start);
SIGTYPE  [R2Q4..R2Q5]  OUT;
SIGTYPE  [R4Q4..R4Q5]  OUT;
      EQUATIONS
           [R2Q4..R2Q5] = [D4..D5] & R2;
           [R4Q4..R4Q5] = [D4..D5] & R4;
      END;
END;

SYM GLB   B1   1   WIDTH, SYNC;
// Video data registers (sync width);
SIGTYPE  [R3Q0..R3Q3]  OUT;
      EQUATIONS
           [R3Q0..R3Q3] = [D0..D3] & R3;
      END;
END;
SYM GLB   B2   1   START, VIDEO 1;
// Video data registers (video start);
SIGTYPE  [R4Q0..R4Q3]  OUT;
      EQUATIONS
           [R4Q0..R4Q3] = [D0..D3] & R4;
      END;
END;

SYM GLB   B3   1   SYNC POSITION CNTR 1;
// Low four bits of sync position counter;
SIGTYPE  [Q0..Q3]  REG OUT;
       EQUATIONS
           [Q0..Q3].CLK = VCLK;
```

```
            Q0  =  (Q0&!VEND)$$((R2Q0&VEND)#(!VEND&!SRST0));
            Q1  =  (Q1&!VEND)$$((R2Q1&VEND)#(!Q0&!VEND&!SRST0));
            Q2  =  (Q2&!VEND)$$((R2Q2&VEND)#(!Q0&!Q1&!VEND&!SRST0));
            Q3  =  (Q3&!VEND)$$((R2Q3&VEND)#(!Q0&!Q1&!Q2&!VEND&!SRST0));
      END;
END;

SYM GLB  B4  1  SYNC POSITION CNTR 2;
// Upper two bits of sync position counter;
// and sync reset signal generation;
SIGTYPE  [Q4..Q5]  REG  OUT;
SIGTYPE  SRST0  OUT;
SIGTYPE  SRST1  REG  OUT;
      EQUATIONS
         Q4.CLK = VCLK;
         SRST1.CLK=VCLK;
            Q4  =  (Q4&!VEND)$$((R2Q4&VEND)#(!Q0&!Q1&!Q2&!Q3&!VEND&!SRST0));
            Q5  =  (Q5&!VEND)$$((R2Q5&VEND)#(!Q0&!Q1&!Q2&!Q3&Q4&!VEND&!SRST0));
          SRST0=!Q0&!Q1&!Q2&!Q3&!Q4&!Q5;
         SRST1.D=SRST0;
      END;
END;

SYM GLB  B5  1  VIDEO START CNTR 1;
// Low four bits of video start counter;
SIGTYPE  [QQ0..QQ3]  REG  OUT;
      EQUATIONS
         [QQ0..QQ3].CLK = VCLK;
          QQ0  =  (QQ0&!SSET)$$((R4Q0&SSET)#(!SSET&!START0));
          QQ1  =  (QQ1&!SSET)$$((R4Q1&SSET)#(!QQ0&!SSET&!START0));
          QQ2  =  (QQ2&!SSET)$$((R4Q2&SSET)#(!QQ0&!QQ1&!SSET&!START0));
          QQ3  =  (QQ3&!SSET)$$((R4Q3&SSET)#(!QQ0&!QQ1&!QQ2&!SSET&!START0));
      END;
END;

SYM GLB  B6  1  VIDEO START CNTR 2;
// Upper four bits of video start counter and;
// START signal generation;
SIGTYPE  [QQ4..QQ5]  REG  OUT;
SIGTYPE  START0  OUT;
SIGTYPE  START1  REG  OUT;
      EQUATIONS
         QQ4.CLK = VCLK;
          START0=!QQ0&!QQ1&!QQ2&!QQ3&!QQ4&!QQ5;
            QQ4  =  (QQ4&!SSET)$$((R4Q4&SSET)#(!QQ0&!QQ1&!QQ2&!QQ3&!SSET&!START0));
            QQ5  =  (QQ5&!SSET)$$((R4Q5&SSET)#(!QQ0&!QQ1&!QQ2&!QQ3&QQ4&!SSET&!START0));
         START1.D=START0
      END;
END;

SYM GLB  B7  1  SYNC WIDTH COUNTER;
// Sync width counter;
```

```
SIGTYPE [QQQ0..QQQ3] REG OUT;
       EQUATIONS
           [QQQ0..QQQ3].CLK = VCLK;
            QQQ0 = (QQQ0&!SRST)$$((R3Q0&SRST)#(!SRST&!SSET0));
            QQQ1 = (QQQ1&!SRST)$$((R3Q1&SRST)#(!QQQ0&!SRST&!SSET0));
            QQQ2 = (QQQ2&!SRST)$$((R3Q2&SRST)#(!QQQ0&!QQQ1&!SRST&!SSET0));
            QQQ3 = (QQQ3&!SRST)$$((R3Q3&SRST)#(!QQQ0&!QQQ1&!QQQ2&!SRST&!SSET0));
       END;
END;

SYM GLB  C1  1  MISC. LOGIC 1;
// Sync width counter SSet signal set-up;
// Sync reset signal generation, video START;
// signal generation;
SIGTYPE SSET0 OUT;
SIGTYPE SSET1 REG OUT;
SIGTYPE SRST OUT;
SIGTYPE START OUT;
       EQUATIONS
           SSET1.CLK=VCLK;
            SSET0=!QQQ0&!QQQ1&!QQQ2&!QQQ3;
           SSET1.D=SSET0;
           SRST=!SRST1&SRST0;
           START=!START1&START0;
       END;
END;

SYM GLB  C2  1  COMPARE, VIDEO END1;
// First eight bits of video end (VEND) comparator;
SIGTYPE VEND1 OUT;
      EQUATIONS
           VEND1 = !((R0Q0$VA0) # (R0Q1$VA1) # (R0Q2$VA2) # (R0Q3$VA3) # (R0Q4$VA4) #
(R0Q5$VA5) # (R0Q6$VA6) # (R0Q7$VA7));
      END;
END;

SYM GLB  C3  1  COMPARE, VIDEO END 2;
// Last two bits of video end (VEND) comparator;
// and VEND signal generation;
SIGTYPE VEND OUT;
      EQUATIONS
           VEND = !((R1Q0$VA8) # (R1Q1$VA9)) & VEND1;
      END;
END;
SYM GLB  C4  1  MEM ADDR MUX 1;
// Video memory address multiplexer bits;
// MA0-MA3;
SIGTYPE [MA0..MA3] OUT;
       EQUATIONS
            [MA0..MA3] = ([A0..A3] & !MUX) # ([VA0..VA3] & MUX);
       END;
END;
```

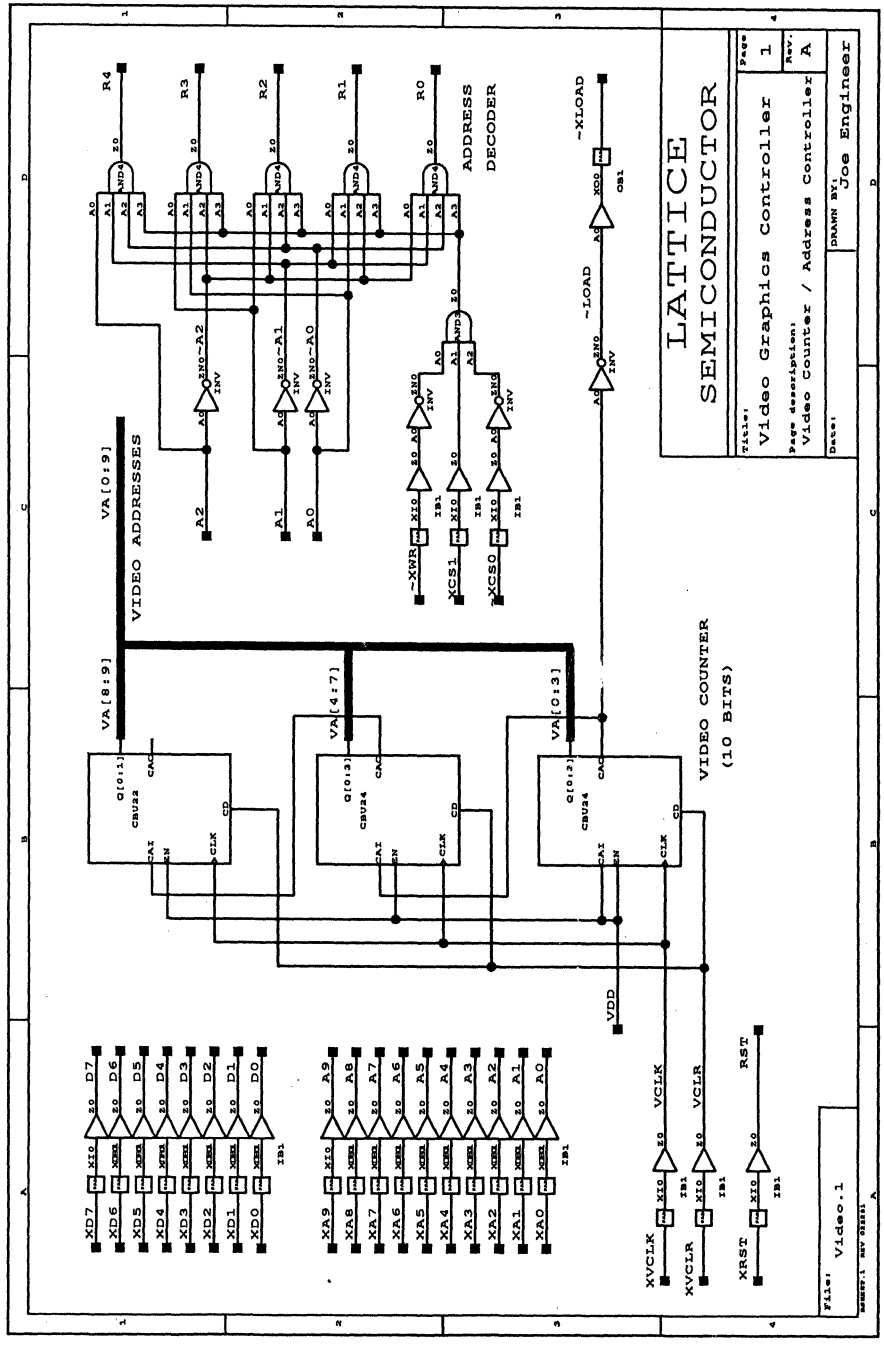
```
SYM GLB  C5  1  MEM ADDR MUX 2;
// Video memory address multiplexer bits;
// MA4-MA7;
SIGTYPE  [MA4..MA7]  OUT;
       EQUATIONS
            [MA4..MA7] = ([A4..A7] & !MUX) # ([VA4..VA7] & MUX);
       END;
END;

SYM GLB  C6  1  MEM ADDR MUX 3;
// Video memory address multiplexer bits;
// MA8 & MA9;
SIGTYPE  [MA8,MA9]  OUT;
       EQUATIONS
            [MA8,MA9] = ([A8,A9] & !MUX) # ([VA8,VA9] & MUX);
       END;
END;

SYM IOC  IO21  1  ;
// Read/Write control signal;
XPIN IO XWR        LOCK 48 ;
IB1  (WR,XWR);
END;

SYM IOC  IO20  1  ;
// Active high chip select;
XPIN IO XCS1          LOCK 3 ;
IB1  (CS1,XCS1);
END;

SYM IOC  IO19  1  ;
// Active low chip select;
XPIN IO XCS0          LOCK 4 ;
IB1  (CS0,XCS0);
END;

SYM IOC  IO0  1  ;
// Address bus A0;
XPIN IO XA0           LOCK 14 ;
IB1  (A0,XA0);
END;
SYM IOC  IO1  1  ;
// Address bus A1;
XPIN IO XA1           LOCK 72 ;
IB1  (A1,XA1);
END;

SYM IOC  IO2  1  ;
// Address bus A2;
XPIN IO XA2           LOCK 15 ;
IB1  (A2,XA2);
END;
```

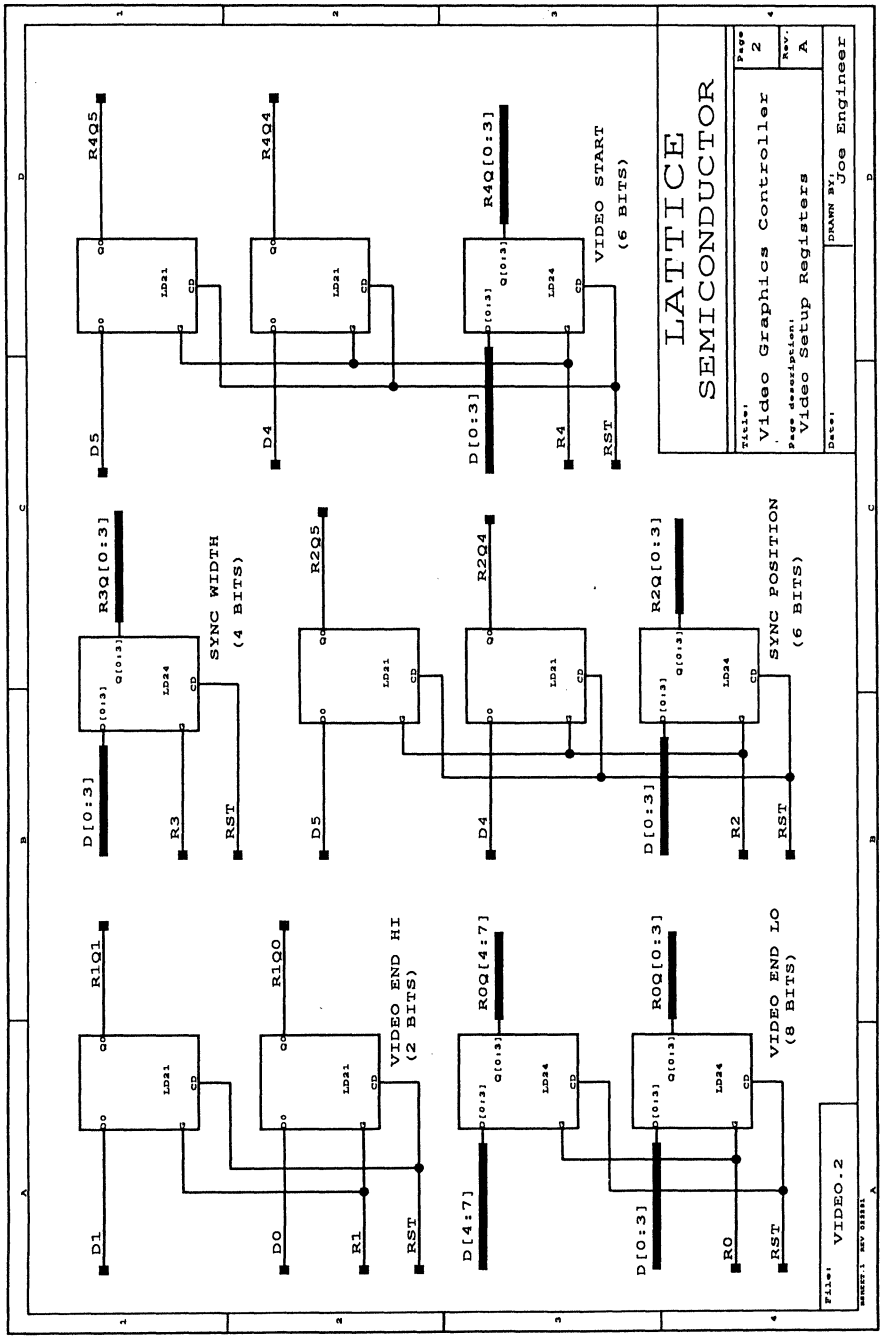
```
SYM IOC  IO3  1  ;
// Address bus A3;
XPIN IO XA3            LOCK 71 ;
IB1  (A3,XA3);
END;

SYM IOC  IO4  1  ;
// Address bus A4;
XPIN IO XA4        LOCK 16 ;
IB1  (A4,XA4);
END;

SYM IOC  IO5  1  ;
// Address bus A5;
XPIN IO XA5            LOCK 70 ;
IB1  (A5,XA5);
END;

SYM IOC  IO6  1  ;
// Address bus A6;
XPIN IO XA6            LOCK 17 ;
IB1  (A6,XA6);
END;

SYM IOC  IO7  1  ;
// Address bus A7;
XPIN IO XA7            LOCK 69 ;
IB1  (A7,XA7);
END;

SYM IOC  IO8  1  ;
// Address bus A8;
XPIN IO XA8            LOCK 18 ;
IB1  (A8,XA8);
END;

SYM IOC  IO9  1  ;
// Address bus A9;
XPIN IO XA9            LOCK 68 ;
IB1  (A9,XA9);
END;

SYM IOC  IO10  1  ;
// Data bus D0;
XPIN IO XD0            LOCK 26 ;
IB1  (D0,XD0);
END;

SYM IOC  IO11  1  ;
// Data bus D1;
XPIN IO XD1            LOCK 60 ;
IB1  (D1,XD1);
```

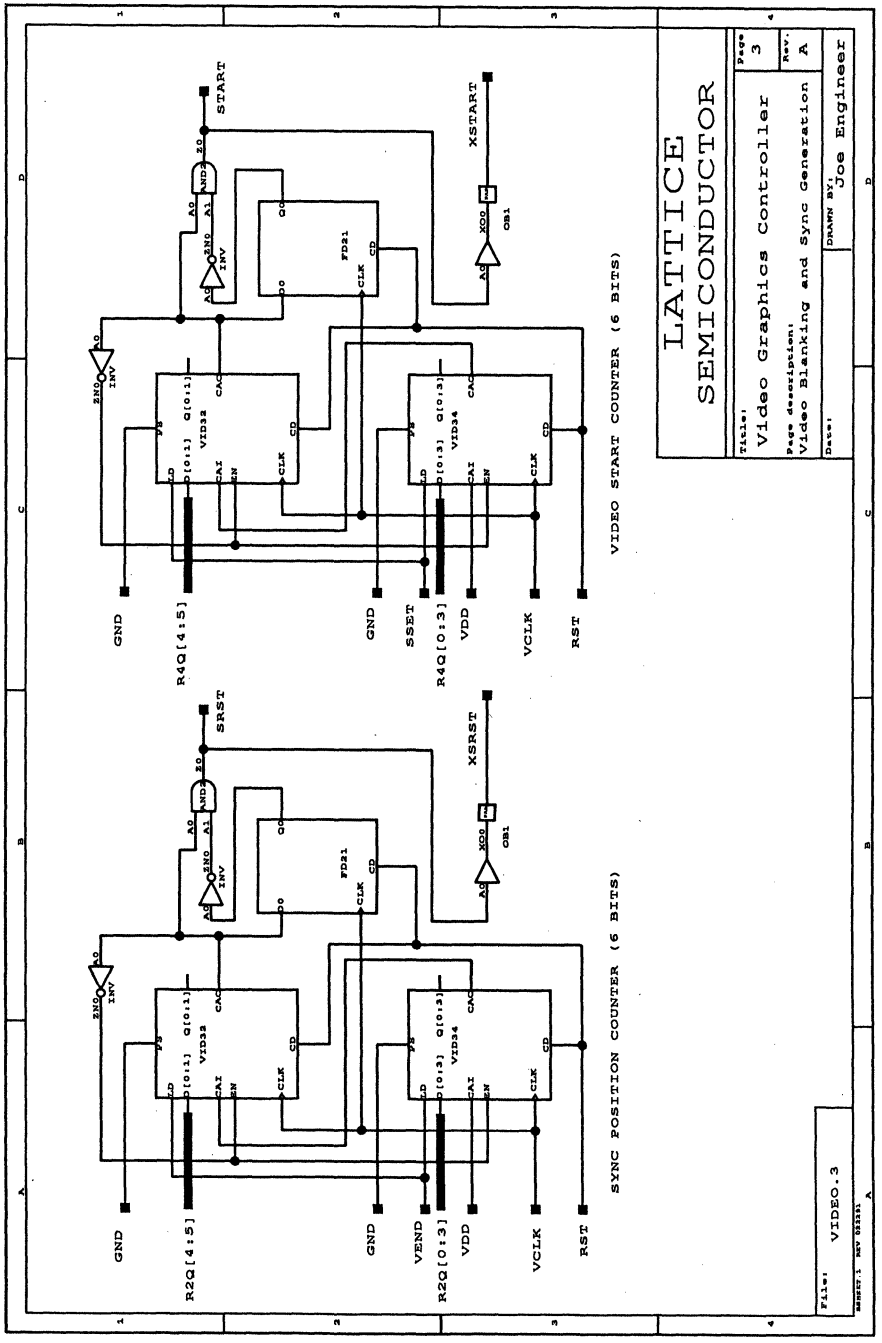
```
END;

SYM IOC  IO12  1  ;
// Data bus D2;
XPIN IO XD2            LOCK 27 ;
IB1  (D2,XD2);
END;

SYM IOC  IO13  1  ;
// Data bus D3;
XPIN IO XD3            LOCK 59 ;
IB1  (D3,XD3);
END;

SYM IOC  IO14  1  ;
// Data bus D4;
XPIN IO XD4            LOCK 28 ;
IB1  (D4,XD4);
END;

SYM IOC  IO15  1  ;
// Data bus D5;
XPIN IO XD5            LOCK 58 ;
IB1  (D5,XD5);
END;

SYM IOC  IO16  1  ;
// Data bus D6;
XPIN IO XD6            LOCK 29 ;
IB1  (D6,XD6);
END;

SYM IOC  IO17  1  ;
// Data bus D7;
XPIN IO XD7            LOCK 57 ;
IB1  (D7,XD7);
END;

SYM IOC  IO18  1  ;
// Video memory address multiplexer;
XPIN IO XMUX        LOCK 55 ;
IB1  (MUX,XMUX);
END;

SYM IOC  IO25  1  ;
// Video memory address MA0;
XPIN IO XMA0           LOCK 49 ;
OB1  (XMA0,MA0);
END;

SYM IOC  IO26  1  ;
// Video memory address MA1;
```

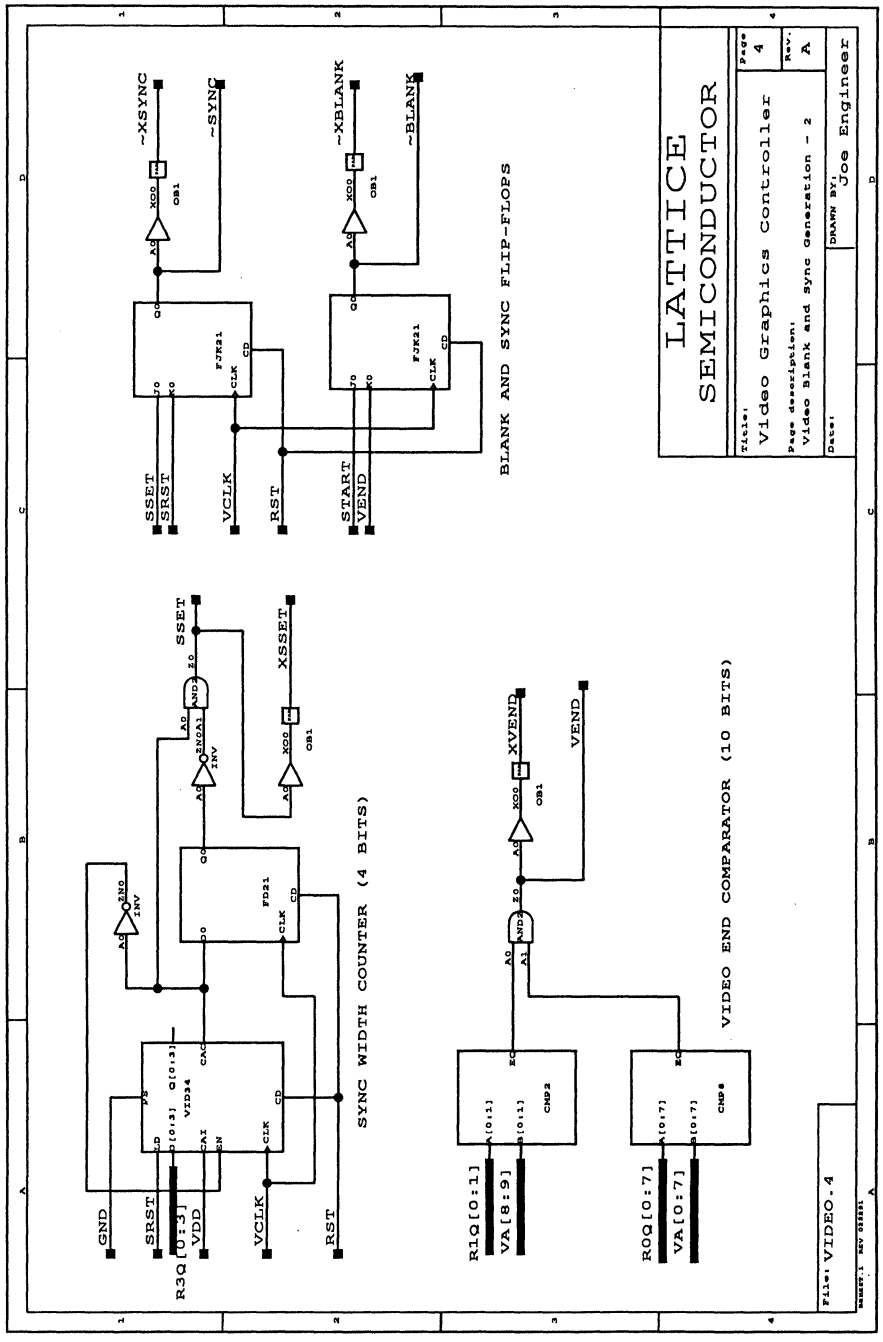
```
XPIN IO XMA1            LOCK 79 ;
OB1  (XMA1,MA1);
END;

SYM IOC  IO27  1  ;
// Video memory address MA2;
XPIN IO XMA2            LOCK 50 ;
OB1  (XMA2,MA2);
END;

SYM IOC  IO28  1  ;
// Video memory address MA3;
XPIN IO XMA3            LOCK 78 ;
OB1  (XMA3,MA3);
END;

SYM IOC  IO29  1  ;
// Video memory address MA4;
XPIN IO XMA4            LOCK 51 ;
OB1  (XMA4,MA4);
END;

SYM IOC  IO30  1  ;
// Video memory address MA5;
XPIN IO XMA5            LOCK 77 ;
OB1  (XMA5,MA5);
END;

SYM IOC  IO31  1  ;
// Video memory address MA6;
XPIN IO XMA6            LOCK 52 ;
OB1  (XMA6,MA6);
END;

SYM IOC  IO32  1  ;
// Video memory address MA7;
XPIN IO XMA7            LOCK 76 ;
OB1  (XMA7,MA7);
END;

SYM IOC  IO33  1  ;
// Video memory address MA8;
XPIN IO XMA8            LOCK 53 ;
OB1  (XMA8,MA8);
END;

SYM IOC  IO34  1  ;
// Video memory address MA9;
XPIN IO XMA9            LOCK 75 ;
OB1  (XMA9,MA9);
END;

SYM IOC  IO35  1  ;
```

```
// Video end output signal;
XPIN IO XVEND        LOCK 45 ;
OB1  (XVEND,VEND);
END;

SYM IOC  IO36  1  ;
// Video sync reset signal;
XPIN IO XSRST        LOCK 46 ;
OB1  (XSRST,SRST);
END;

SYM IOC  IO37  1  ;
// Video sync width set output signal;
XPIN IO XSSET        LOCK 30 ;
OB1  (XSSET,SSET);
END;

SYM IOC  IO38  1  ;
// Video start output signal;
XPIN IO XSTART       LOCK 47 ;
OB1  (XSTART,START);
END;

SYM IOC  IO39  1  ;
// Video load output signal;
XPIN IO XLOAD      LOCK 32 ;
OB1  (!XLOAD,!LOAD);
END;

SYM IOC  IO40  1  ;
// Video Blanking output signal;
XPIN IO XBLANK      LOCK 36 ;
OB1  (XBLANK,BLANK);
END;

SYM IOC  IO41  1  ;
// Video sync output signal;
XPIN IO XSYNC        LOCK 31 ;
OB1  (XSYNC,SYNC);
END;

SYM IOC  Y0  1  ;
// Video clock input signal;
XPIN CLK XVCLK          LOCK 20 ;
IB1  (VCLK,XVCLK);
END;

SYM IOC  I0  1  ;
// Video counter clear input;
XPIN I XVCLR        LOCK 25 ;
IB1  (VCLR,XVCLR);
END;
END;   //LDF DESIGNLDF
```

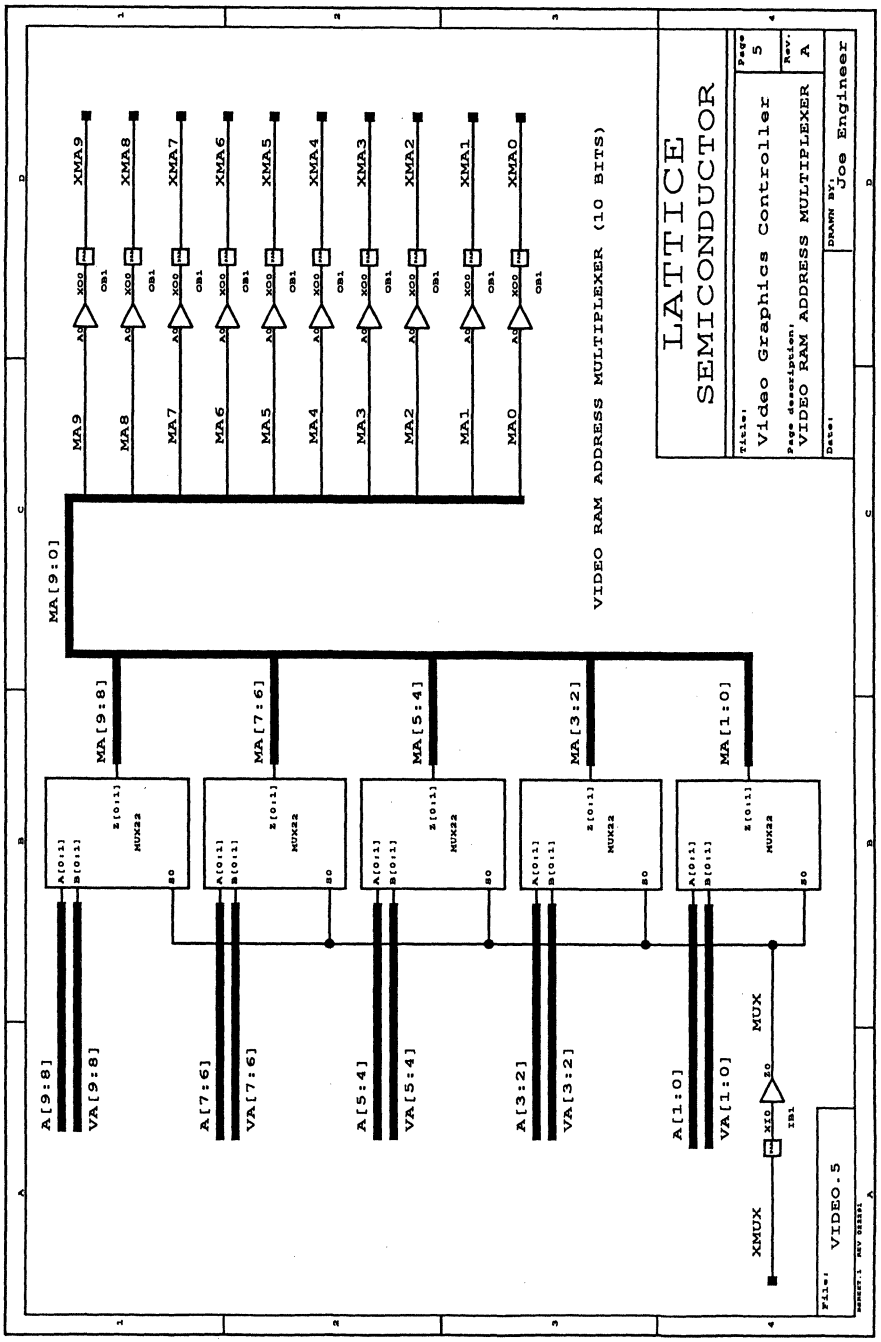

LATTICE
SEMICONDUCTOR
Video Graphics Controller
Video Counter / Address Controller
Joe Engineer
ADDRESS
DECODER
VIDEO ADDRESSES
VIDEO COUNTER
(10 BITS)
~XLOAD
~XWR
XCS1
XCS0
VDD
XVCLK
XVCLR
XRST
VCLK
VCLR
RST

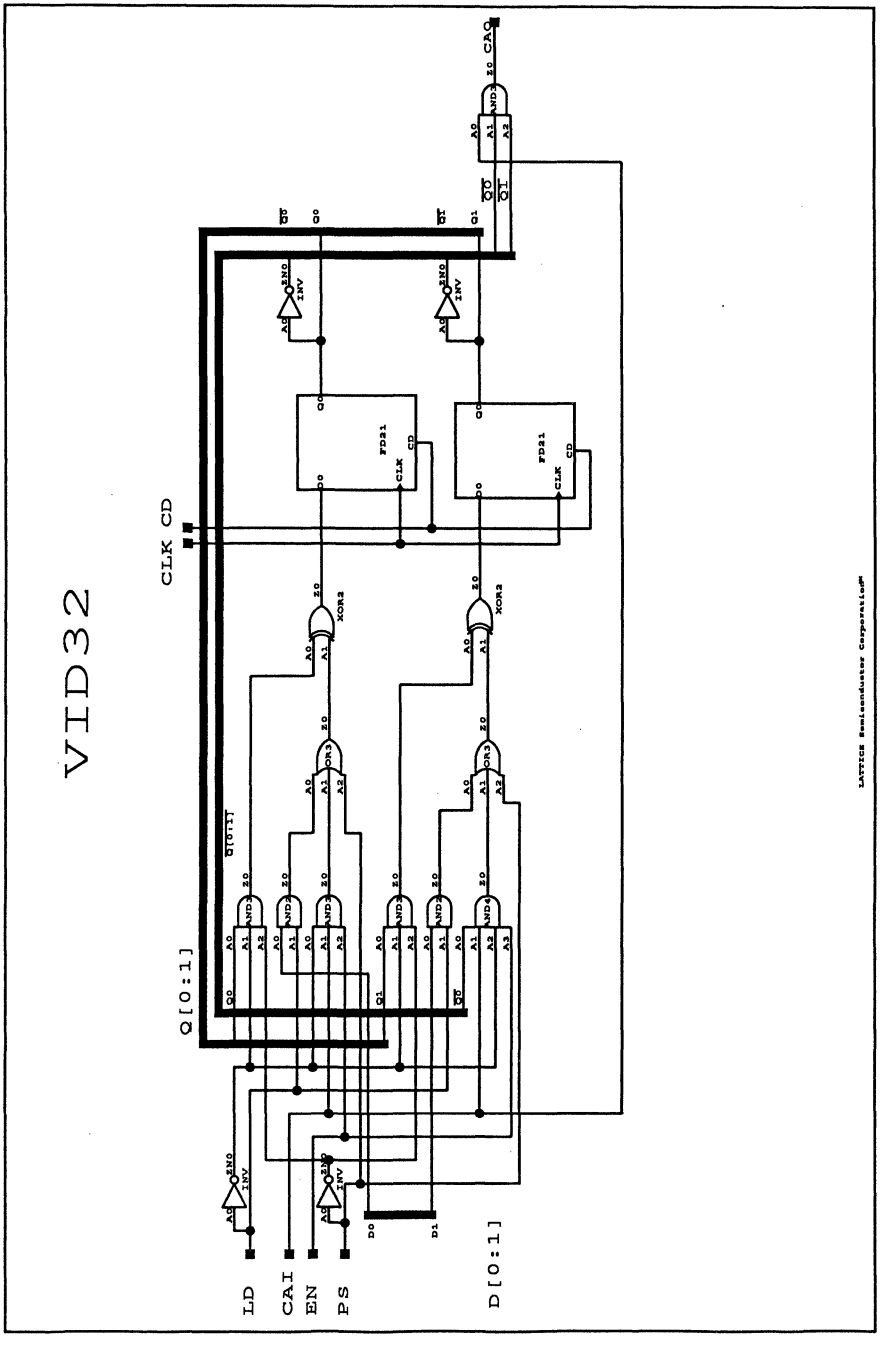

LATTICE
SEMICONDUCTOR
Title: Video Graphics Controller
Page description: Video Setup Registers
Page 2
Rev. A
DRAWN BY: Joe Engineer
Date:
File: VIDEO.2
R4Q5
R4Q4
R4Q[0:3]
VIDEO START (6 BITS)
D5
D4
D[0:3]
R4
RST
R3Q[0:3]
SYNC WIDTH (4 BITS)
R2Q5
R2Q4
R2Q[0:3]
SYNC POSITION (6 BITS)
D[0:3]
R3
RST
D5
D4
D[0:3]
R2
RST
R1Q1
R1Q0
VIDEO END HI (2 BITS)
R0Q[4:7]
R0Q[0:3]
VIDEO END LO (8 BITS)
D1
D0
R1
RST
D[4:7]
D[0:3]
R0
RST

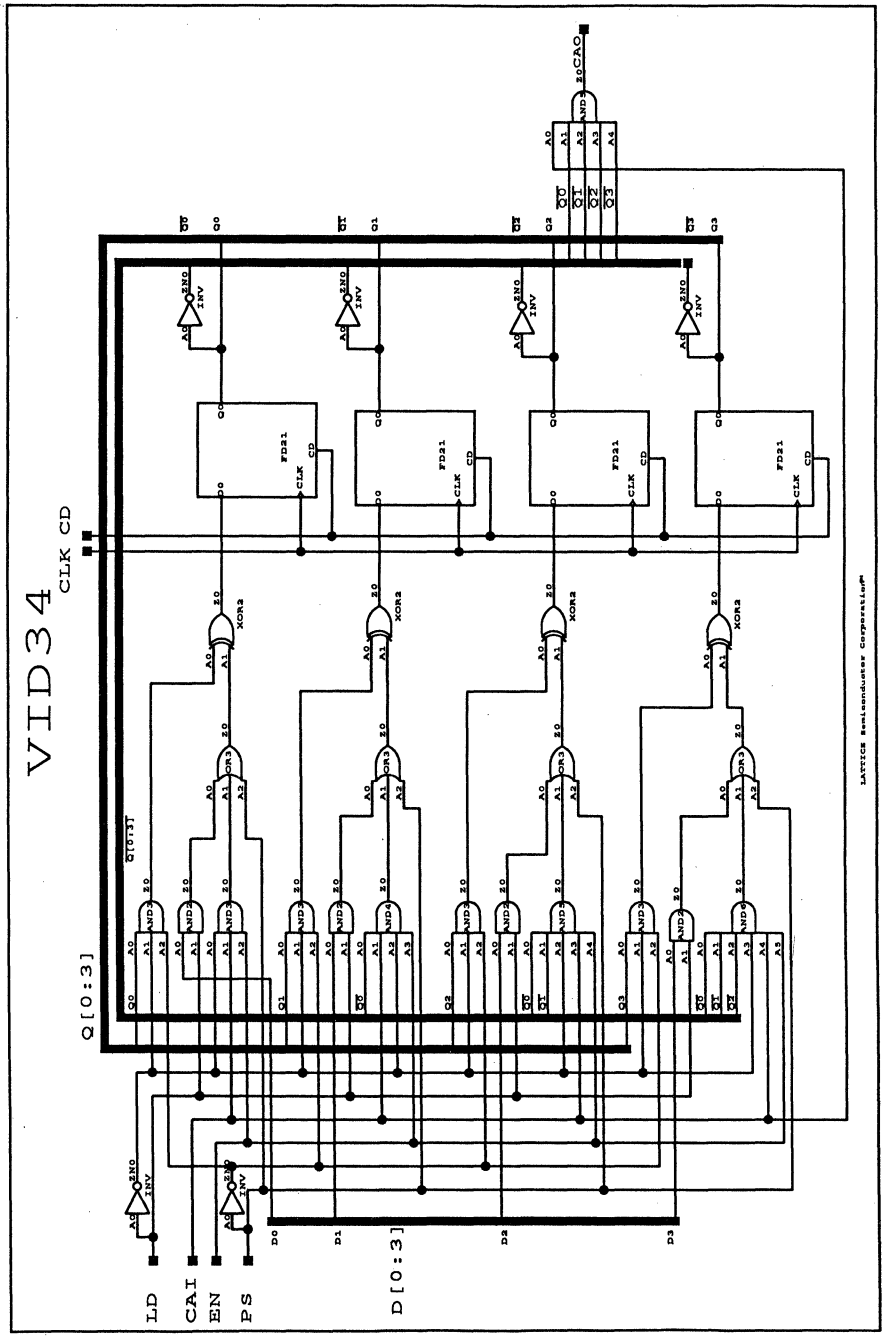

LATTICE
SEMICONDUCTOR
Title: Video Graphics Controller
Page 3
Rev. A
Page description: Video Blanking and Sync Generation
Date:
DRAWN BY: Joe Engineer
File: VIDEO.3
VIDEO START COUNTER (6 BITS)
SYNC POSITION COUNTER (6 BITS)
START
XSTART
SRST
XSRST
GND
SSET
VDD
VCLK
RST
VEND
R4Q[4:5]
R4Q[0:3]
R2Q[4:5]
R2Q[0:3]


LATTICE
SEMICONDUCTOR
Title:
Video Graphics Controller
Page description:
Video Blank and Sync Generation - 2
Date:
Page
4
Rev.
A
DRAWN BY:
Joe Engineer
File: VIDEO.4
~XSYNC
~SYNC
~XBLANK
~BLANK
SSET
SRST
VCLK
RST
START
VEND
BLANK AND SYNC FLIP-FLOPS
SYNC WIDTH COUNTER (4 BITS)
VIDEO END COMPARATOR (10 BITS)
XSSET
XVEND
VEND
GND
R3Q[0:3]
R1Q[0:1]
VA[8:9]
R0Q[0:7]
VA[0:7]

8


LATTICE SEMICONDUCTOR
Title: Video Graphics Controller
Page description: VIDEO RAM ADDRESS MULTIPLEXER
Date:
DRAWN BY: Joe Engineer
Page 5
Rev. A
VIDEO RAM ADDRESS MULTIPLEXER (10 BITS)
MA[9:0]
MA9
MA8
MA7
MA6
MA5
MA4
MA3
MA2
MA1
MA0
XMA9
XMA8
XMA7
XMA6
XMA5
XMA4
XMA3
XMA2
XMA1
XMA0
MA[9:8]
MA[7:6]
MA[5:4]
MA[3:2]
MA[1:0]
MUX22
A[9:8]
VA[9:8]
A[7:6]
VA[7:6]
A[5:4]
VA[5:4]
A[3:2]
VA[3:2]
A[1:0]
VA[1:0]
MUX
XMUX
File: VIDEO.5


VID32
CLK CD
Q[0:1]
D[0:1]
LD
CAI
EN
PS
CAO
FD21
XOR2
IND2
IND3
IND4
INV
LATTICE Semiconductor Corporation


VID34
CLK CD
Q[0:3]
D[0:3]
LD
CAI
EN
PS
CAO
Q0
Q1
Q2
Q3
D0
D1
D2
D3
FD21
XOR2
XOR3
AND2
AND3
AND4
INV

Lattice

Compiling Multiple PLDs into a pLSI Device

January 1992 | Application Note 2506

Introduction

As high density programmable devices become more complex, they can combine larger designs previously implemented with low density PLDs and SSI/MSI glue logic. The use of pLSI™ devices from Lattice Semiconductor can reduce manufacturing costs by: shrinking board size, simplifying test procedures, speeding development, and reducing the type and number of parts required to be kept in inventory. Designers familiar with PLDs and SSI/MSI devices can convert to Lattice pLSI devices with little effort. This application note addresses a procedure to convert a circuit designed with PLDs, MSI, and SSI devices into the Lattice Semiconductor pLSI device format.

The basic steps required to convert the design are:

- ❑ Define the I/Os
- ❑ Convert the Low Density PLD Equations
- ❑ Combine the PLD Source Files
- ❑ Add any MSI, SSI Functions
- ❑ Partition the Logic into Generic Logic Blocks (GLBs)
- ❑ Import the File into the pLSI Design Environment
- ❑ Place and Route Using the pLSI Development System (pDS™)

Define the I/Os

The first task in the conversion process is to define the I/O pins of the Lattice pLSI device based on the circuit developed using lower density devices. One must determine if the design is I/O limited or gate limited. If the design is I/O limited the circuit must be partitioned into a higher pin count device, or two (or more) lower pin count devices. A gate limited design will mandate the design be partitioned into a higher density pLSI device. This implies that there will be unused I/O pins. This can allow additional functionality to be designed into the Lattice pLSI device, providing the device does not become gate limited again.

A straightforward approach to estimate gate count is to use SSI, MSI and PLD equivalents. By adding up the total number of these circuit blocks required for a circuit, one can determine if the design will fit into a Lattice pLSI device. For example, the GLB (Generic Logic Block) of the Lattice pLSI family has 18 inputs and 4 outputs. Numerous functions implemented in 16V8, 20V8 and 22V10 devices can be fit easily into one GLB. However, in cases where five or more outputs are desired, partitioning into 2 GLBs will be necessary. Expanding this analogy, approximately 1 MSI device and 2 SSI devices can fit into a single GLB.

When converting a circuit implemented with MSI, SSI and PLDs, partitioning can be achieved by recognizing which nodes are best suited for interconnection within the pLSI device. The partitioning of logic will vary for different MSI, SSI or PLD devices. By determining which of these devices will be implemented completely within the Lattice pLSI device, it will become readily apparent which of the nodes should be kept within the pLSI device or allocated as an I/O pin. Signals which connect to a device not implemented within the Lattice pLSI device will be required to be an I/O. As a shot gun approach, one can simply draw a box around the circuit, count the I/O and gate requirement, and select the pLSI device meeting the requisite gate and I/O count. This task requires good engineering judgement and knowledge of device architecture to effectively utilize the pLSI device architecture.

Nodes which have a broad fanout should be considered for I/O unless all destination devices are implemented within the Lattice pLSI device. Naturally, nodes going off-board must be implemented as I/O pins on the Lattice pLSI device.

Clocking is another factor to consider when partitioning a circuit. If the circuit requires more than the four global clocks available in the pLSI device, the circuit should be partitioned so that circuits with common clocks are in the same pLSI device. The global clock inputs are available on pins Y0, Y1, Y2 and Y3. Y0, Y1 and Y2 can be directly connected to any GLB, while Y2 and Y3 can be directly connected to any I/O cell. Alternatively, each GLB can generate its own clock from the output of a single product term within the GLB. This will allow up to 32 separate clocks within the pLSI 1032.

8

LATTICE SEMICONDUCTOR CORP., 5555 Northeast Moore Ct., Hillsboro, Oregon 97124, U.S.A.
Tel. 1-800-LATTICE (528-8423); FAX (503) 681-3037

PLD File Conversion

Once the circuit to be placed into the Lattice pLSI device has been defined, the process of converting the design into the pLSI format begins. Typically a design will be implemented with PLDs and a small number of MSI and SSI devices. Most of the PLD devices will have an associated source equation file. This file can be used as the basis for the design equations to be imported into the Lattice pDS software.

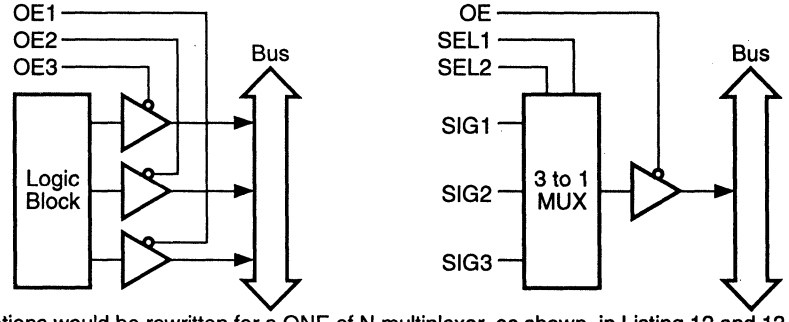
Adding MSI and SSI Functions

By creating Boolean equations which emulate an SSI or MSI function, and subsequently importing that file into the Lattice pDS software, SSI and MSI functions can be easily integrated into the design. Another method of implementing these functions is to look through the Lattice pDS software Macro Library (or the pLSI and ispLSI Software Manual), to find the closest equivalent circuit to the function desired. This Macro can then be edited if necessary, to provide the exact function required. The net result of either of these processes is to derive functionally correct equations which best utilize the pLSI device architecture.

Conversion of 3-Stated to Multiplexed Signals

Internal 3-state buses implemented in an ASIC or high density PLD can create problems such as undefined outputs. A better implementation of internal 3-state functions is to implement them with a ONE of N multiplexer function. The inputs to the multiplexer are the signals that are 3-stated together. The select lines of the multiplexer are individual 3-state enable signals. This technique is commonly used in the design of ASICs. The block diagram of figure 8-45 illustrates the idea.

Figure 8-45. Block Diagram of a Multiplexer Emulating a 3-State Buffer



The 3-state equations would be rewritten for a ONE of N multiplexer as shown in Listing 12 and 13.

Listing 12. Original 3-State Equations

```
BUSA_SIG1= SIG_1A              BUSB_SIG1= SIG_1B
BUSA_SIG1.OE=SIG1_OE           BUSB_SIG1.OE=!SIG1_OE
BUSA_SIG2= SIG_2A              BUSB_SIG2=SIG_2B
BUSA_SIG2.OE=SIG2_OE           BUSB_SIG2.OE=!SIG2_OE
BUSA_SIG3= SIG_3A              BUSB_SIG3=SIG_3B
BUSA_SIG3.OE=SIG3_OE           BUSB_SIG.OE=!SIG3.OE*
```

Listing 13. Multiplexer Equations

```
BUSA_OUT=  (!SIG1_OE  &  !SIG2_OE)  &  SIG_1A
         # ( SIG1_OE  &  !SIG2_OE)  &  SIG_2A
         # ( SIG1_OE  &   SIG2_OE)  &  SIG_3A;

BUSB_OUT=  (!SIG1_OE  &  !SIG2_OE)  &  SIG_1B
         # ( SIG1_OE  &  !SIG2_OE)  &  SIG_2B
         # ( SIG1_OE  &   SIG2_OE)  &  SIG_3B;
```

The original equations have been implemented as BUSA_OUT and BUSB_OUT. The six original equation sets have been reduced to two.
*Note that SIG3_OE is not needed in this implementation.

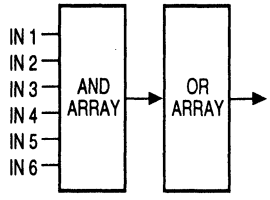
The AND function of the output enables (SIG1_OE, SIG2_OE) does not increase the number of product terms required to implement the various bus signal functions. This will always be true for product term oriented architectures such as the Lattice pLSI family of devices. There are ten ONE of N multiplexer Macros currently available in the pLSI Macro Library. By using these Macros, the conversion may be readily accomplished by simply changing the default signal names within the Lattice Macro.

Inversion Placement

High density Programmable Logic device architectures vary greatly, active high versus active low internal signals may provide a significant savings in the utilization of the Lattice pLSI device resources as shown in the following example (see figures 8-46 and 8-47).

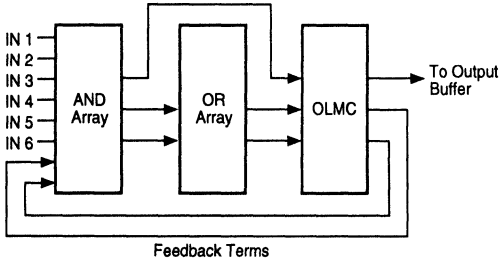
Consider the equation shown in Listing 14.

Figure 8-46. Implementation of Active Low Equation



If this equation cannot be implemented with a hardware inverter, Boolean expansion will produce the equation as listed in listing 15.

Figure 8-47. Implementation of Active High Equation



Listing 14.

```
!OUT = IN1 & IN2 & IN3 # IN4 & IN5 & IN6;
```

Listing 15.

```
OUT= ! IN1 #! IN2 #! IN3 &! IN4 #! IN5 #! IN6;
```

The second equation requires 7 product terms, as opposed to 2 when implemented into the pLSI architecture as shown in figure 8-48. Notice that the boolean expansion also requires two extra feedback terms. When manipulating equations to fit the Lattice pLSI device architecture, consider placing inversions for active low outputs at the signal destination or at the I/O cell. The Lattice pLSI family can accommodate any active low signal with this technique as all inputs to the logic block have both true and complementary inputs. In other words a signal "A" routed to a GLB, will have both "A" and "!A" available within the GLB AND array. The outputs of the Lattice pLSI devices can also be selected as active high or active low.

Defining a Preset/Reset Mechanism

A frequently neglected but necessary requirement is a reset mechanism. All state machine designs should have a known power up state. If a reset line is routed to all state machine registers for reset, significant routing resources will be unnecessarily used. The reset mechanism should take advantage of the hardware reset resources available in the Lattice pLSI device. Individual reset signals should be removed from the design equations and the hardware reset should be used. The Lattice pLSI family has two reset mechanisms. A global reset for all registers and an asynchronous reset for each generic logic block (GLB) or I/O cell.

Many high density device architectures provide only reset and no preset mechanism. Consider complementing the output requiring preset and using the hardware reset. If that is not possible, make the preset synchronous by adding a preset term into the design equations.

If new logic is to be placed in the high density device, partition the logic into available logic blocks or GLBs. Simply enter the Boolean Equations or use the available Macros. With the exceptions of a few key words, the equations are entered similar to those of DATA I/O's ABEL™ system.

Circuit Partitioning

The *.DOC files produced by third party compilers are in an industry standard format. These files contain the reduced equations which are derived from the source file, JEDEC maps, high level state machine language, truth table, or standard equations. The individual PLD and SSI/MSI *.DOC files should be combined into a single source file for partitioning into the pLSI device.

By grouping the equations into groups of no more than four outputs, the PLD equations can be partitioned to fit into the GLBs of the pLSI device since there are 4 outputs per GLB. Headers and trailers must be placed around the four equations to indicate to the Lattice pDS software, into which GLB the equations should be loaded. The syntax is shown in table 8-5.

Each GLB has 18 inputs, 20 product terms and 4 registered or combinatorial outputs. Additionally, there is product term combining among the four outputs and an optional Exclusive OR gate which is fed by a single product term and an AND/OR term. The software will automatically place a given set of four equations into a GLB.

If the PLD equations do not fit into a GLB, the Lattice pDS software will give a message as to why. If there are too many inputs, the equation can be moved into another GLB and a new equation brought into the current GLB which does not exceed the limit of 18 inputs.

As previously stated, every GLB is allowed one clock. This clock may come from either one of the four global clocks or a clock generated from a product term (.PTCLK). Ensure all registered outputs in a GLB have a single clock.

If an equation contains product terms which cannot be allocated into one GLB, consider exchanging a complex equation for one of less complexity in another GLB. If this trading of equations is not possible, simply move the equation into an empty GLB. In general, try to keep equations with common inputs in the same GLB. If a function requires a high number of product terms (product term combining), try to make use of the product term groups as illustrated in figure 8-48.

Moving a registered equation from one GLB to the next will not degrade performance as the interconnect delays between all GLBs are constant. Combinatorial equations may have an extra GLB and unit interconnect delay added to the propagation delay - if the implementation requires more than 18 inputs and 20 product terms. If an equation will not partition into a single GLB, the equation must be split into two equations and then cascaded. For Registered equations consider pipelining the intermediate equation(s) to keep the performance at the same level.

The above steps are all that are required to place PLD type designs into the GLBs of the Lattice pLSI family. Note that no syntax changes of the AND/OR portions of the equations were required.

Definition of I/O Cells

The final step in the conversion process is to define the I/O cells. The I/O cell definition is shown in listing 17. Because the device is routed according to signal names, all I/O cells will automatically be connected to the proper internal nodes.

Import and Verify the Design

Now that the design has been partitioned into the Lattice pLSI GLBs, the device source file needs to be imported into the Lattice pDS software so that it can be verified, placed, and routed. By using the FILE and IMPORT.LDF commands, the text file containing the design will be imported into the Lattice pDS software. The pDS software will check the syntax of each GLB and I/O cell as the file is imported.

Table 8-5. Header and Trailer Syntax

Header	PLD Equations	Trailer
SYM GLB A0 <GLB NAME> 1;	Signal1.clk=.....;	END;
SIGTYPE Signal1 REG OUT;	Signal1=.....;	END;
SIGTYPE Signal2 OUT;	Signal2=.....;	
SIGTYPE Signal3 CRITICAL OUT;	Signal3=.....;	
SIGTYPE Signal4 REG OUT;	Signal4=.....;	
EQUATIONS		

All syntax errors must be eliminated to successfully import a file. After the converted low density PLD logic has been imported into the Lattice pDS software, any SSI or MSI devices can be placed into GLBs. This can be done by using the available Macros from the Lattice Macro Library, or using Boolean Equations. After this is accomplished perform a DESIGN VERIFY. This command performs a global design rule check. After a successful global verify the design is ready for automatic Place and Route.

Place and Route is invoked with the DESIGN ROUTE commands. Lattice specifies 100% "push button" routing with 80% GLB usage. After place and route the fusemap can be generated and the design downloaded to a programmer. In the case of an isp (in-system programmable) device, the device can be programmed via the download cable connected to a parallel port of an IBM compatible PC.

As can be seen from the information contained in this technical note, converting a design from low density PLDs to the Lattice pLSI family of high density programmable logic is quick and easy as long as a few guidelines are followed:

1) Decide if the Design is I/O or Gate Limited.

2) Choose the Appropriate Lattice pLSI Device.

3) Use as Much of the Original Boolean Equations From the Low Density Source File as is Practical.

4) Convert 3-state Outputs to a ONE of N Multiplexer Scheme.

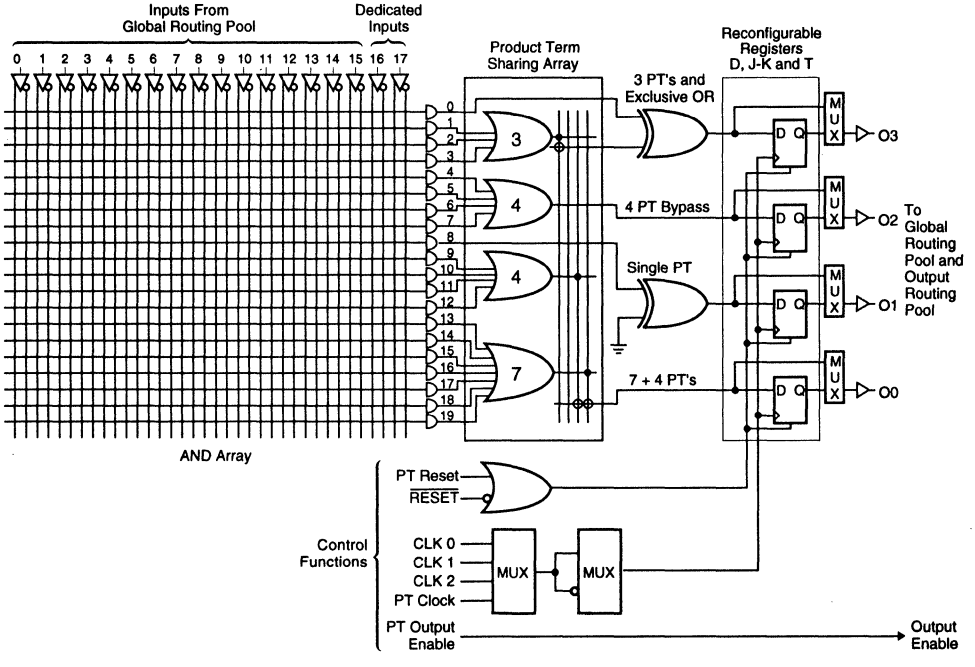
5) For Reset Functions, use the Global Reset for the Entire Device or the Asynchronous Reset for Specific GLBs.

6) Use No More than 18 Inputs or 4 Outputs per GLB When Partitioning the Logic.

Listing 17. I/O Cell Definition

```
SYM IOC IOXX 1;
 XPIN XSIGNAME PIN# LOCK#;
   IB1/OB1  (SIGNAMEIN/SIGNAMEOUT,SIGNAMEIN/SIGNAMEOUT);
END;
```

Figure 8-48. GLB Simplified Diagram Showing Product Term Sharing Combinations



Section 1: Introduction to pLSI™ and ispLSI™

Section 2: pLSI and ispLSI Data Sheets

Section 3: GAL® Data Specifications

Section 4: pLSI and ispLSI Architecture

Section 5: pLSI and ispLSI Advantages

Section 6: pLSI and ispLSI Software Development Tools

Section 7: pLSI and ispLSI Design Optimization

Section 8: pLSI and ispLSI Application Notes

Section 9: pLSI and ispLSI Article Reprints

Fast, High-Density PLDs Offer On-Board Reprogramming 9-1

Lattice Fields FPGA 9-3

Avoid The Pitfalls Of High-Speed Logic Design 9-5

Multiple Factors Define True Cost of PLDs 9-11

Section 10: pLSI and ispLSI Quality, Reliability & Programmer Support

Section 11: Sales Information

Fast, high-density PLDs offer on-board reprogramming

Designers hunting for high density and performance in a programmable logic device may find what they're looking for in the pLSI and ispLSI, two high-density PLD families from Lattice Semiconductor. Delivering 70-MHz system speed, the devices combine the high performance and ease of use of PLDs with the density and flexibility advantages of field-programmable gate arrays. And the ispLSI family are the first high-density PLDs to offer nonvolatile, in-system programming.

There are eight devices in the pLSI and ispLSI families, four of which offer Lattice's proprietary in-system programming (isp) technology. System performance of 70 MHz is achieved with a total delay, from input to output, of 15 ns.

Except for a few inputs on the ispLSI devices (for controlling the isp function), the pLSI and ispLSI have the same internal arrangement. Their architecture consists of multiple generic logic blocks (GLBs), each of which implements 90 percent of all 4-bit MSI functions. A proprietary routing network provides global interconnect, 100 percent routability and over 80 percent device utilization. Devices range in pin count from 44 to 120 (with 32 to 104 I/Os). The equivalent PLD gate counts of the series range from 2,000 to 8,000 gates, according to Lattice.

Interconnect routing

At the heart of the pLSI's architecture is an interconnect routing method, which is handled by a global routing pool (GRP). The GRP is responsible for data transfers from either the inputs or the I/O of the device to a GLB, or from one GLB to another.

Key to overall device performance, the GRP takes only a few nanoseconds to perform such transactions. Unlike other routing structures, the GRP has a brief, predictable delay time. This delay, when added to a logic block's delay, results in a fixed total delay of 15 ns. This level of predictability isn't possible in FPGAs, where delay times are often dependent upon the location of the GLB.

The advantage of a predictable transaction speed between logic blocks is twofold. First, knowing the delay times ahead of time reduces the need for testing at every design iteration and, therefore, improves time-to-market. Second, the task of optimizing the speed paths in the design is less complex. With a set delay of 15 ns, it's easier for the simulation software to compile a design. Designers won't have to wait as long while their logic compilers crunch through their place-and-route schemes, optimizing high-speed paths.

pLSI and ispLSI at a glance

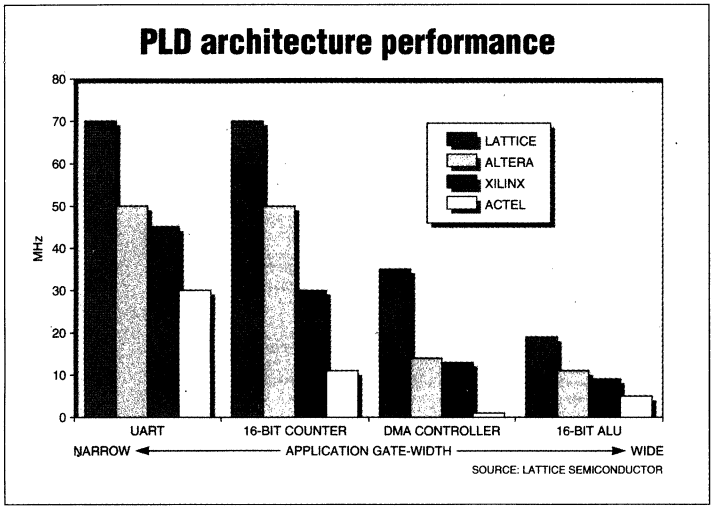
- First in-system programmable PLDs
- 15-ns input-to-output delay
- 2,000 to 8,000 equivalent PLD gates
- Multiple generic logic block architecture

On-board reprogramming

While otherwise identical to the pLSI devices, the ispLSI devices are the first electrically erasable and in-system programmable PLDs on the market. These features let the devices be programmed, reprogrammed, or reconfigured for test without being removed from the circuit board. They also let system designers perform real-time prototyping and debug. And they aid end users by letting the products they own be reconfigured or upgraded on the spot.

Through mode control, four of the dedicated inputs on the ispLSI can be turned into in-system programmable functions. Using only TTL-level programming inputs, the functions can be programmed, reprogrammed or erased, even when the device is installed on a board or multichip module.

Users can program part of the device or the whole device at once. The danger of accidentally repro-



According to tests conducted by Lattice Semiconductor, the new pLSI and ispLSI high-density PLD families operate at 70 MHz in narrow-gate logic functions, such as UARTs (universal asynchronous receiver/transmitters) and 16-bit counters. In wide-gate logic structures, such as DMA controllers and 16-bit ALUs, the Lattice architecture remains above 20 MHz.

9

gramming the device is small because accessing those pins essentially requires state-machine programming. This provides a sufficiently complex combination lock to make accidental reprogramming unlikely.

Software support for the pLSI families consists of entry-level and advanced software packages. The entry-level package, which runs on PC compatibles, provides for Boolean and macro input. A second package for more-advanced users runs on PC compatibles or workstations and adds schematic capture, and logic- and timing-simulation tools.

Available now, the first of the pLSI family is the pLSI 32-70. This 84-pin device has 32 GLBs, 64 I/O pins and 192 registers. In a PLCC, the part is priced at $98.50 (100s). The first of the ispLSI devices will begin shipping 4Q91. —*Jeffrey Child*

Lattice Semiconductor
5555 NE Moore Ct.
Hillsboro, OR 97124
1-800-327-8425

Electronic Engineering
TIMES

Monday
June 10, 1991
Issue 645

A CMP Publication

THE INDUSTRY NEWSPAPER FOR ENGINEERS AND TECHNICAL MANAGEMENT

Uses E²PROM PROGRAMMING

Lattice fields FPGA

By STAN BAKER

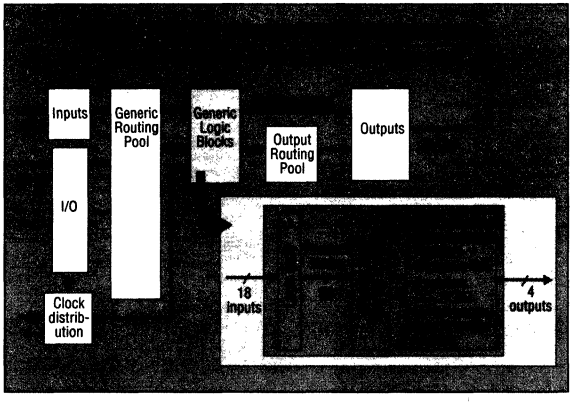
Hillsboro, Ore. - Lattice Semiconductor Corp. is jumping into high-complexity PLDs with two families of devices that compete with FPGAs. The move significantly broadens the company's thrust in the logic market and puts it in the middle of the fastest-growing market segment. The new products also make Lattice the first company to move up from PAL devices to FPGAs with its own architecture.

Lattice's new pLSI and ispLSI devices will offer up to 8,000 equivalent gates. Both families will use E²PROM programming for the first time in FPGAs, and the ispLSI family will be in-system programmable.

"I see the '90s as the decade of programmability," said Cyrus Tsui, president and CEO of Lattice. Market-research firms suggest he's right. They indicate the bipolar portion of the PLD business is dropping. Meanwhile, the CMOS portion is growing at a compound annual growth rate of more than 40 percent, and the high-complexity segment of the CMOS market is the most active.

"From a global standpoint, entering the high-density market will double Lattice's total addressable market from 1 billion to 2 billion dollars in 1994," Tsui said. "In practical terms, these products will enable us to reach a class of customers in telecommunications, industrial control and the military that we currently cannot serve with our low-density GAL product offerings. This is both a unique product and company positioning."

The new architecture features flexible granularity-the circuit blocks provide highly flexible programming of product terms, flexible choice of product-term sharing and steering, and choices of four outputs from the combinatorial and registered ports of the block.



Four functions

A single circuit block can implement up to four functions, since it has four outputs and the logic-array flexibility to separate them in the programmed cell routing. It can also implement functions from two to 18 lines wide in a single block, offering significant delay-time advantages over architectures that have to implement wide functions in a series of logic levels (see related story, on reverse side).

Unlike FPGAs, the design of the routing resources and the architecture of the circuit blocks makes the timing characteristics of these devices pre-

dictable-or "deterministic," to use the industry term-along with those of Altera and Plus Logic. Precise timing can be predicted from data sheets. This differs from the timing of the true FPGAs of Xilinx and Actel, for which the data sheet can give only statistical estimates of actual post-route timing.

Meets requirements

Lattice is coming to market with up to 8,000 equivalent gates soon after Xilinx forecasted 20,000 gates and Toshiba projected 40,000 gates for next year. But Lattice's complexity meets the current requirements of most customers for FPGAs and even gate arrays. As for the future path for the new architecture, "we plan to expand to include mask versions of these devices for high-volume designs," said Steve Donovan, director of marketing at Lattice. "And current product plans call for densities exceeding 20,000 gates. We will introduce these devices as market conditions warrant."

Lattice specifies its pLSI devices to work at system clock rates up to at least 70 MHz. That's for a single circuit block operating through an input and output circuit. The delay time for such a function is 15 ns.

Lattice claims the circuit modules, called generic logic blocks, can implement 90 percent of all 4-bit MSI functions. The proprietary routing network provides global interconnectivity, 100 percent routability and over 80 percent device utilization.

The new Lattice products encompass eight devices, four of which offer in-system programming. The pinout count ranges from 44 to 120. The number of I/Os range from 32 to 104.

The ispLSI types are programmed, reprogrammed, and reconfigured for test without having to be removed from the circuit board. Real-time prototyping and debugging, and reconfiguring or upgrading the system in the field, are possible. With the E²PROM programming technology, the devices are guaranteed to deliver 100 percent programming yields and 100 percent conformation to functional, ac and dc specifications.

Same process

The 0.8-micron process that is used to fabricate these devices is developed and maintained by Lattice engineers. The UltraMOSIV process is the same used in the 7.5-ns and 10-ns GAL devices now in production at Lattice.

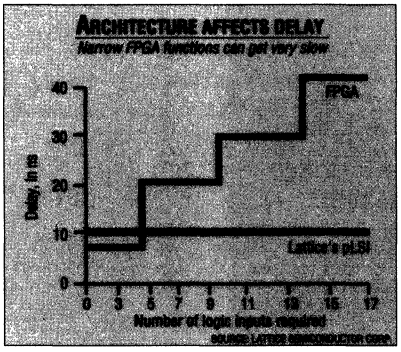
The first pLSI devices will begin shipping in the third quarter of this year. The in-system programming ispLSI versions will come a quarter later. Software and programming support will accompany the first product shipments.

Beta testing of the new pLSI devices will begin next month. Engineering samples will be generally available by October. The general market release of software and silicon will come in November. The price of the 6,000-gate device with 192 registers in an 84-pin PLCC will be $98.50 each in lots of 100.

Lattice aims at best of architectural worlds

FPGA architectures generally implement gating functions up to four or five lines wide. Wider functions are then implemented using a series of such narrow functions. But every level of logic costs more delay time. Wider functions save time, but they also can cost silicon area and waste circuitry where the width is not needed.

Lattice Semiconductor is trying to offer the best of both worlds with logic blocks that can handle widths up to 18 signals, but do so in logic arrays within the blocks that can be used for several narrow-width functions when the wide ones are not needed.



However, the company's research shows the most popular widths for gating functions are from eight to 12 lines. The current FPGA and other complex PLD devices offer less width than what is most needed, requiring two or three logic levels and, consequently, two or three circuit delays. The accompanying figure, developed by Lattice engineers, shows the pLSI devices with a 10-ns delay per block not changing its delay from two through 18 inputs.

But a competing FPGA with five-input blocks starts at 8 ns, but adds another 8 ns each time the width expands by five inputs. For the most popular widths, from eight to 12, the competing FPGAs will have block delays of 20 ns to 30 ns. The delays are in 10-ns increments because of interconnect delays beyond the 8-ns delay in each block.

For wide gating applications, such as large counters, wide address decoding and multiplexing, the system speed slows considerably in current FPGAs because of the the need to cascade the logic block outputs.

—Stan Baker

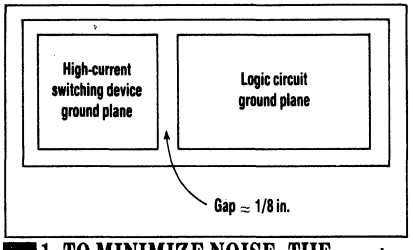
MAKE SURE THAT YOUR TURBO-CHARGED LOGIC SYSTEM WORKS BY PAYING AS MUCH ATTENTION TO PRINTED-CIRCUIT BOARD LAYOUT TECHNIQUES AS TO LOGIC DESIGN CONSIDERATIONS.

AVOID THE PITFALLS OF HIGH-SPEED LOGIC DESIGN

Modern high-speed systems demand modern high-speed logic families. Consequently, semiconductor houses have developed such product lines as ACT, FACT, and AS. But these systems also demand that the lay-out of their boards conform with the results of distributed-element theory, otherwise ringing, crosstalk, and other transmission-line phenomena render those systems inoperative. Meeting this second requirement necessitates something more than a new product introduction—it insists on a change in the way logic boards are engineered. The logic-systems designer and the board-layout designer must work hand-in-hand if a viable high-speed board or system is to be produced.

In the past, logic design and board layout were usually regarded as separate parts of the design process. First the system designer configured the logic, then the board engineer laid it out. That approach worked because slew rates were so low (0.3 to 0.5 V/ns) that crosstalk wasn't much of a problem; rise times were so long (4 to 6 ns) that ringing could settle down before a logic element could change state; and in general, the assumptions of lumped-element circuit theory usually worked out pretty well.

For systems designed with today's high-speed logic circuitry, those underlying assumptions no longer hold true. Today's slew rates are on the order of 2 to 3 V/ns, rise times are below 2 ns (frequently, below 1 ns), and transmission-line phenomena, such as ringing, can be a problem for trace



1. TO MINIMIZE NOISE, THE ground plane should be fragmented into separate areas for noisy high-current devices and for sensitive logic circuits. For best results, the number of signal lines that cross the gap between the fragments should be minimized.

JOCK TOMLINSON
Lattice Semiconductor Corp., P.O. Box 2500, Portland, OR 97208; (503) 681-0118.

Reprinted with permission from **ELECTRONIC DESIGN** - November 9, 1989

DESIGNING WITH HIGH-SPEED LOGIC

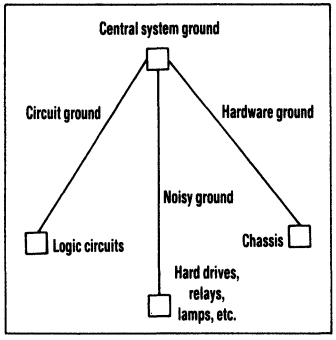
lengths as short as 7 in. As a result, logic designers must take certain steps:

- Use ground and power planes.
- Control conductor spacings to eliminate crosstalk.
- Make extensive use of decoupling capacitors.
- Pay attention to ac loading.
- Terminate lines properly to minimize reflections.

PLANE ADVICE

For high-speed logic, ground planes aren't simply suggested for reliable board performance—they are absolutely necessary. It's essential that one layer of the board be assigned for a ground plane and that it cover as large an area as possible. A solid ground plane lowers the ground-return-path impedance as well as the device-to-device ground pin impedance.

But a common ground plane for all of the circuitry in a system can cause problems by coupling noise from high-current switching devices into sensitive logic inputs. Therefore, the ground plane for such high-current devices as relays, lamps, motors, and hard drives should be separated from the logic ground. This can be accomplished by fragmenting the ground plane into discrete areas *(Fig. 1)*.



2. SEPARATE DEDICATED grounds should be supplied for the logic circuitry, noisy high-current devices, and the chassis. The three should come together at one point, the central system ground, which is usually located near the power supply.

But fragmentation causes problems of its own—it creates discontinuities in the characteristic impedance of any transmission line that crosses the separation between fragments. Therefore, for best results, boards should be laid out so that only two fragments are needed. The gap between those fragments should be kept as narrow as possible (an eighth of an inch works well in most applications), and the number of signal lines that cross the gap should be minimized. Designers should also bear in mind that through-holes and vias subtract from the effective area of the plane, increasing its effective impedance.

As with grounding, an entire layer of the board should be designated as a power plane. Even though it is at a different potential, the power plane should be implemented in accordance with the same concepts as the ground plane. Therefore, it should be fragmented when necessary to isolate noisy components from delicate logic circuits.

A WELL-GROUNDED SYSTEM

In addition to properly designed power and ground planes, high-speed logic systems require the establishment of a good, clean (low-

SIGNAL LINES BECOME TRANSMISSION LINES

For the transmission line model illustrated in the diagram, the rise time (t_R) is less than the line propagation delay (T_D). In other words, a complete TTL level transition will occur before the pulse is received at the receiving end of the line and reflections (ringing) will result. The voltage change at point A on the line is expressed in Eq. 1:

$\Delta V_A = \Delta V_{int} (Z_0 / (R_0 + Z_0))$

Where: V_{int} = internal voltage on the output of the driver;

R_0 = output impedance of the driving gate;

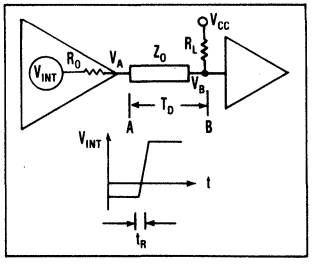
R_L = load impedance;

Z_0 = the characteristic line impedance;

and V_A = the source voltage at the sending end of the line.

Because R_0 is so small when compared to the line impedance, the change in voltage at point A (ΔV_A) will approximately equal the change in internal voltage (ΔV_{int}). This voltage transition propagates down the line and is seen at point B after the line propagation delay, T_D.

At point B, a portion of the wave will be reflected back towards point A in accordance with the formula (Eq. 2):



Eq. 2

$\rho_L = (R_L - Z_0) / (R_L + Z_0)$

where ρ_L, called the voltage reflection coefficient (rho), is the ratio of the reflected voltage to the incident voltage.

After examining Eq. 2, it should be evident that $-1 \leq \rho \leq +1$. It should also be evident that there will be no reflected wave if $R_L = Z_0$—if the line is terminated in its characteristic impedance. Note that the reflected wave can, in principle, be as large as the incident voltage and of either positive or negative polarity.

This analysis holds true for the sending end of the line, as well as the receiving end. That is,

Eq. 3

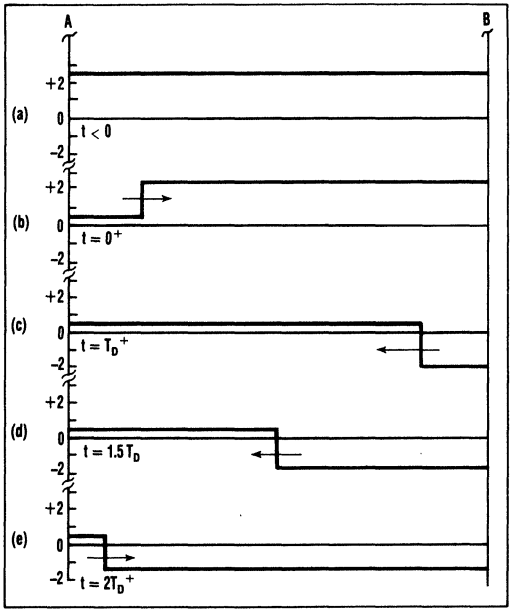
$\rho_S = (R_0 - Z_0) / (R_0 + Z_0)$

DESIGNING WITH HIGH-SPEED LOGIC

noise) system ground for reliable performance. A clean system ground ensures less noise within the system, and thus ensures good, strong transistor margins. At least 10% of the ground connections on the pc card should be connected to the system ground to reduce card-to-ground impedance.

Like the ground and power planes of the individual boards, the overall grounding scheme should be fragmented with separate conductors provided for the various sections of the system. For example, all relays, lamps, hard drives, and other noise-generating devices should have their own separate ground path. The system's mechanical package (chassis, panels, and cabinet doors) should have a dedicated ground. And, of course, the logic circuitry should have a ground of its own.

Those three grounds should then come together at the central system ground point, which will usually be located near the power supply *(Fig. 2)*. This common-point grounding technique can also be very effective in reducing radiated interference (EMI and RFI).



3. WAVE PROPAGATION along a transmission line occurs as follows: Prior to time zero, there is a steady-state voltage of 2.5 V dc on the line (a). At t = 0, the voltage at point A drops to 0.5 V, sending a negative pulse of -2 V toward point B (b). At t = T_D, that negative pulse is reflected from point B. It adds algebraically to the 0.5 V on the line and sends a -1.5-V pulse back toward point A (c). The reflections then continue as in (d) and (e).

TAMING CROSSTALK

Crosstalk—the undesirable coupling of a signal on one conductor to one on a nearby conductor—becomes an increasingly serious problem as slew rates go up. This signal coupling is made worse if the second trace has a high impedance or if the traces run parallel to one another for more than a few inches and are spaced less than 100 to 150 mils apart.

Crosstalk can be catastrophic to a logic board, sabotaging a conceptually flawless piece of logic design. For example, if a clock line and a data line run parallel to each other for more than several inches, and if the data line cross-couples or superimposes its signal onto the clock line, the device that the clock is driving may detect an illegal level transition.

Methods to reduce crosstalk are straightforward, though not particularly elegant. The coupling can be attenuated by separating the adjacent traces as much as possible. The trouble with this approach is that available board real estate often limits the possible separation to an inadequate amount.

Ground striping, or shielding, is an effective way to reduce crosstalk and it makes better use of available board area. With ground striping, a ground trace (the stripe) is run between the two parallel traces to act as a shield. If ground striping is used, through holes to the ground plane should be placed every 1 to 1.5 inches along the ground strip to eliminate the possibility of inadvertently creating a stub or a high-frequency antenna.

Another step that can be taken to reduce crosstalk is to lower the impedance of those traces into which crosstalk is especially to be avoided. The lower the impedance that a trace presents, the harder it will be to cross-couple a signal into it.

Even with the use of power and ground planes on a pc board, decoupling capacitors must be used on the V_{CC} pins of every high-speed device. Those devices demand a nearly instantaneous change in current whenever they switch states. Because the power plane can't meet that demand, a high-quality decoupling capacitor is required, otherwise the switching will cause noise on the V_{CC} plane.

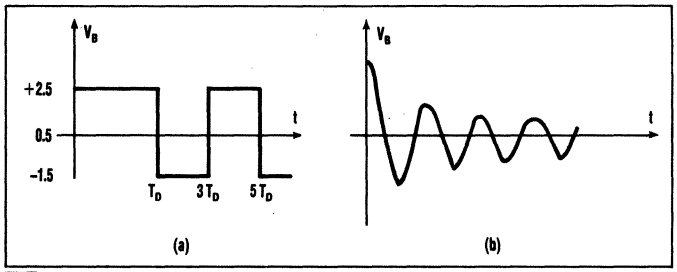
A 0.1-μF multilayer ceramic (MLC) or other RF quality (low-inductance) capacitor should be placed on every fast-slew-rate device as close to the V_{CC} pin as possible. The commercially available DIP sockets with built-in decoupling capacitors also work well in this application.

Most designers, when they think of loading at all, think in terms of dc loading—traditionally referred to as fan-out and fan-in. But that type of loading rarely presents a problem with today's state-of-the-art logic devices. Much more significant when designing with high-speed logic are input and output ac loading.

INPUT CAPACITANCE

Because the input capacitance of a device impacts the overall performance of the logic circuit, it should be examined before a particular device is selected for a design. To ensure specified performance, the total load capacitance that a device drives—including the distributed ca-

DESIGNING WITH HIGH-SPEED LOGIC



4. IDEALLY, THE VOLTAGE at point B oscillates forever between +2.5 V and -1.5 V (a). In reality, it will be a damped ringing (b).

pacitance of the trace—shouldn't exceed the device's specified capacitive load. Most high-speed logic devices have a maximum loading of 50 pF. As a rule of thumb, the maximum load on any logic element should be no more than four to six devices for best speed/load performance. However, there are some high-slew-rate devices on the market that have higher output drive capabilities.

BEWARE OF AUTOROUTER

The most common reason for not following the board-layout principles mentioned so far is having an autorouter do the layout. Autorouters do what they were designed to do very well: They place traces so as to make the most efficient use of the pc-board real estate. But most autorouters don't have the capability to determine which devices are high-speed and which are not. This is where the logic designer must step in and lay out sections, or islands, of high-speed logic by hand in order to avoid the pitfalls of designing with high-speed logic.

TRANSMISSION LINES

In addition to the common-sense layout considerations discussed so far, designers of high-speed systems must have at least a basic understanding of transmission lines and proper termination techniques *(see "Signal Lines Become Transmission Lines," p. 76)*. The reason: As frequencies go up, wavelengths come down to the point where they are of the same order as circuit-board dimensions. Once that happens, any connection between devices should be considered a transmission line. The lumped-element assumption is simply invalid above that point.

The most common consequence of failing to consider the distributed nature of a high-speed logic board is ringing, which is caused by multiple reflections from the ends of unterminated transmission lines. An unterminated line has no load impedance ($R_L = \infty$) and is therefore an impedance-mismatched line. The behavior of this line when connected to a device with a fast slew rate can be understood from the following example: Prior to time zero, there's a steady-state voltage of 2.5 V dc at all points on the line *(Fig. 3a)*. At t = 0, an initial TTL voltage transition from 2.5 V to 0.5 V occurs at point A *(Fig. 3b)*. Time T_D later, the signal reaches point B and is reflected by the load reflection coefficient, ρ_L.

The input impedance of the device at point B is very high with respect to Z_O; R_L can be approximated by infinity. By plugging into Eq. 2 from the box *(p. 76)*, the reflection coefficient approximately equals +1. In other words, the voltage reflected by the load is equal to the incident voltage *(Fig. 3c)*. The reflected wave passes back along the signal path toward point A *(Fig. 3d)*.

Repeating the calculations for the sending end of the line (point A), where $R_O \approx 0$, you get a value for the source reflection coefficient, ρ_S, of −1. In other words, there are reflections from the source as well as the load, but the source reflects the inversion of the wave that is incident upon it *(Fig. 3e)*.

Looking just at the behavior of the signal at point B, the single-step volt-

RULES TO REMEMBER

The following ten rules summarize everything the logic designer needs to know when designing with high-speed CMOS.

1) Keep signal interconnections as short as possible.

2) Use a multilayer PCB.

3) Provide ground and power planes. Discontinuities in the planes should be avoided because reflections can occur from abrupt changes in the characteristic impedance.

4) Fragment the ground and power planes to supply separate sections for high-current switching devices.

5) Use decoupling capacitors on every high-speed logic device (0.1 μF MLC type) located as close to the V_{CC} pin as possible.

6) Provide the maximum possible spacing among all high-speed parallel signal leads.

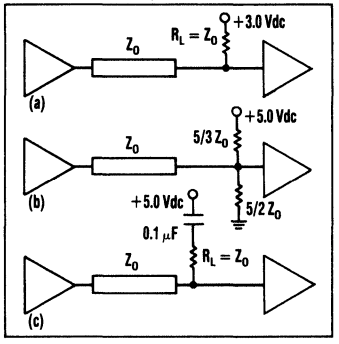
7) Terminate high-speed signal lines where $t_R < 2T_D$.

8) Beware of ac loading conditions within the design. Exceeding the manufacturer's recommended operating conditions, especially for capacitance, can cause problems.

9) When using parallel termination, put bends in all high-speed signal runs that go to more than one load. Use a termination load at the absolute end of the line.

10) Create islands of high-speed devices on the pc board. This simplifies board layout and ropes-off the high-speed areas.

DESIGNING WITH HIGH-SPEED LOGIC



5. THE BASIC PARALLEL termination scheme works well but requires a separate 3-V supply (a). The Thevenin equivalent eliminates the need for a separate supply, but dissipates extra power from the regular 5-V supply (b). The use of a capacitor cuts dc dissipation altogether while supplying ac termination (c).

age transition at t = 0 leads to an endlessly oscillating signal with a total voltage swing of 4.0 V—twice the original level transition. The voltage doubling comes about because the voltage at point B is the sum of the incident and reflected waves at that point *(Fig. 4a)*. Actually, because of the non-ideal nature of a real circuit board (finite input and output impedances, losses in the transmission lines, and so forth), ρ_L will be less than +1, and ρ_S will be greater than −1. As a result, the reflections will become successively smaller, causing the familiar damped ringing condition *(Fig. 4b)*.

If the ringing amplitude is large enough, it can cause the receiving device to see an illegal level transition and possibly result in spurious logic states occupying the logic design. In some cases, the amplitude of the ringing can actually be large enough to damage the input of the receiving device.

TERMINATE YOUR TROUBLES

The way to eliminate ringing on a transmission line is to terminate the line in its characteristic impedance at either the sending or receiving end. The most common way to terminate a line is with a parallel termination at the receiving end *(Fig. 5)*.

In the configuration *(Fig. 5a)*, $R_L = Z_0$ and R_L is pulled up to 3 V dc. In principle, R_L could be tied to ground, but TTL-compatible devices could not then supply the necessary drive.

Solving for ρ_L *(Eq. 2)*, it can be seen that $\rho_L = 0$. Terminating a line in its characteristic impedance results in a reflection coefficient of zero, which means that there will be no reflections or distortions on the line. Other than the time delay, T_D, the line will act as if it were a dc circuit. It's important to note that even though devices or gates may be placed at any location on the line, the terminating resistor should be placed at the end of the line. In no case should the line be split like a Tee to feed several devices in parallel *(Fig. 6a)*. Instead, it should be serpentined to feed them sequentially *(Fig. 6b)*.

The 3-V power source shown *(Fig. 5a)* appears at first to be a major drawback, but R_L and the power supply can be expressed as a Thevenin equivalent running off the system power supply of 5 V dc *(Fig. 5b)*. This variant works well, but the designer should bear in mind that it dissipates additional power.

REDUCING DISSIPATION

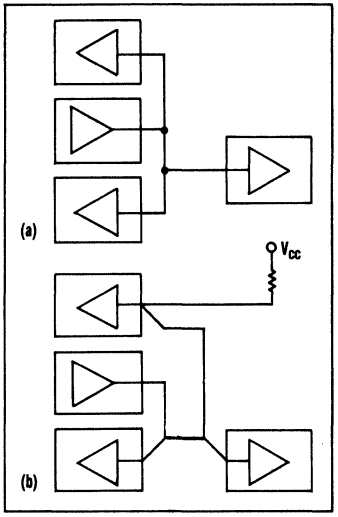
A solution that dissipates less power than either of the others uses a capacitor to cut the dc dissipation to zero *(Fig. 5c)*. The recommended capacitor is a 0.1-μF MLC type. Several manufacturers produce both capacitor-resistor and pull-up/pull-down termination packs. The pull-up/pull-down packs usually come in a single in-line package (SIP) with pins on 0.1-in. centers, while the capacitor-resistor combination comes in a standard 16-pin DIP. The most common SIP pull-up/pull-down resistor values are 220Ω/330Ω, 330Ω/470Ω combinations.

An alternative to a parallel termination at the receiving end is a series termination at the sending end *(Fig. 7)*. The idea behind serial termination is to make $\rho_S = 0$ and $\rho_L = +1$. To do so, R_L is made equal to infinity (left unterminated) and a series resistor is added at the source to make the overall source impedance equal to the characteristic impedance of the line—that is, $R_S + R_0 = Z_{OL}$.

Making $R_S + R_0$ equal to Z_{OL}, of course, creates a voltage divider, which puts half of the signal amplitude across the line and half across the series combination of R_S and R_0. Therefore, with the series termination, the amplitude of the transmitted wave is half of what it would be without the termination.

Interestingly enough, the unterminated receiving end of the line precisely compensates for this halving of the amplitude. The reason is as follows: At the receiving end, the half-amplitude wave is received and a half-amplitude wave is reflected. But bear in mind that those are two separate waves whose amplitudes add at the point of reflection. As a result of this addition, the only thing seen at the receiving end of the line is a full-size pulse.

The main disadvantage of a series termination is that the receiving gate or gates must be at the end of the line—no distributed loading is possible. The obvious advantage of a series termination over a parallel one is that a series termination doesn't



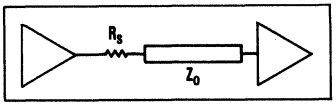
6. SERPENTINING IS essential when terminating a line. Never split the line to feed parallel devices (a). Rather, feed them sequentially with a serpentined line (b).

DESIGNING WITH HIGH-SPEED LOGIC

require any connection to a power supply.

Transmission-line effects must be taken into consideration whenever line propagation delays get up to the point where a signal transition can be completed before that signal can travel down a line, be reflected, and travel back to its starting point. In other words, lines must be terminated when,

$2T_D = T_R$.



7. THE SERIES termination needs no pull-up supply. Its main disadvantage is that it can't handle distributed loads.

CALCULATING DELAY

Taking 2 ns as a typical rise time for a state-of-the-art high-speed logic device, how long can a board trace get before its propagation delay gets to be 1-ns long? For a pc board with a continuous ground plane and a signal trace on the adjacent layer, the propagation delay depends on only one variable, the dielectric constant of the board material. That delay time is given by:

$$t_{PD} = 1.017\,(0.475\,e_R + 0.67)^{1/2}\ \text{ns/ft}$$

For a typical board constructed of FR4 material, e_R (the dielectric constant) is 4.7 to 4.9. If an average e_R of 4.8 is used in the equation, then t_{PD} turns out to be 1.75 ns/ft, which works out to 6.86 in./ns. As a rule of thumb, then, any line that is over 7 in. long should be considered a transmission line and approached accordingly.□

Jock Tomlinson, senior applications engineer at Lattice, holds a BSEE from Colorado State University.

Multiple factors define true cost of PLDs

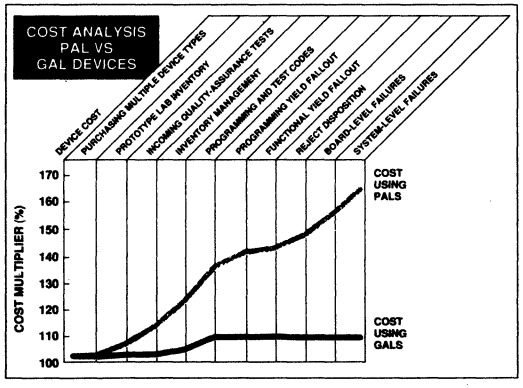
By DEAN SUHR

Designers using PLDs (programmable logic devices) for system design and manufacturing traditionally think of piece price as the key consideration in the PLD selection process. Thanks to recent advances in technology, however, the system cost of using PLDs is influenced by factors such as fabrication technology, device quality, reliability and yield.

System cost is quite different from the sum of the component costs. The price paid for a device or component represents only one part of the system cost of a PLD; the systems team also has to consider the costs hidden in the programming, handling, quality control, throughput and overhead that's necessary to get a "raw" PLD to a functional state on a board.

Because the true system cost of a PLD is the sum of the piece price and all of these hidden costs, and is spread over several functions and departments, it's often difficult to define and measure. In most companies, for example, purchasing and engineering define the parts list and acquire the parts. These departments are often under pressure to reduce absolute unit cost and to purchase the least-expensive part available.

But the profitability of both the product line and the company is based not on device acquisition cost but on total system cost. Buying the least-expensive part may not provide the lowest total system cost.



Programmability can play a significant role in the total cost of PLDs. The example above was taken from a system that used 100,000 bipolar PLDs per year. Moving to E²CMOS GALs can reduce total cost by up to 34 percent.

Hidden cost factors

To calculate the system cost of using a particular PLD type, vendor and technology, managers must also take into consideration the additional costs of purchasing overhead, inventory management, prototype inventory and quality assurance (QA). Purchasing overhead can add 2 percent to the actual device cost. As the number of inventory line items rises, the overhead needed to purchase those items increases. PLDs with generic architectures can minimize the number of different devices a company must purchase, and therefore reduce purchasing costs.

As much as 10 percent of a device's cost can be attributed to inventory management overhead, including shelf space, depreciation, count management, obsolete write-offs, and safety stock. Reducing inventory line items simplifies the management overhead, in turn cutting costs.

BY ELIMINATING YIELD AND HANDLING LOSS, IMPROVING QUALITY, AND SIMPLIFYING INVENTORY MANAGEMENT, DESIGNERS CAN CUT THE TRUE COST OF USING PLDS

PLDs are particularly adaptable to just-in-time (JIT) inventory management systems, which minimize inventory by increasing throughput. Using a JIT system constrains a company's flexibility because the company must carry fewer items. But adding PLDs to the inventory will let the same narrow range of products provide a wide variety of functions.

Macrocell-based PLDs also have increased the flexibility of companies that use them and reduced stocking requirements. In the past, designers using fixed-architecture PLDs had to keep in stock every PLD architecture required for a design. Macrocell-based devices, on the other hand, can be configured to emulate dozens of old architectures and many new configurations.

The macrocell used in E²CMOS generic array logic (GAL) devices goes one step further. These devices also offer 100 percent socket compatibility with older programmable array logic (PAL) architectures, so designers can simply substitute the GAL device for the old PLD architecture. No redesign is necessary. Existing JEDEC files and master devices can be used, reducing system cost.

9

No longer a simple calculation, the system cost of using programmable logic devices is affected by their fabrication technology, testability and impact on inventory management.

The cost of the prototype inventory also influences total system cost. Although engineering labs are stocked with devices for building and debugging prototypes, many companies meet engineering lab shortages by borrowing from manufacturing stock. This policy can shrink manufacturing inventories and, by doing so, can increase the system cost of the remaining manufacturing units by as much as 1 percent when units are ordered to restock the shelves.

All PLDs have a programmable element that determines their functionality and ac/dc performance. These programmable elements can be fabricated from metal-link fuses, programmable diodes or transistors, volatile static RAM cells, UV EPROM cells or EEPROM cells. Each of these technologies varies in programmability and has a different impact on device performance and reliability.

Each programmable element also offers a different erase capability. Metal-link and one-time-programmable devices, for instance, can't be erased. UV EPROM devices can be erased, but this process requires an expensive windowed package and takes 20 to 30 minutes. EEPROM devices offer instant erasability in as little as 50 ms. Technologies that aren't erasable or that have lengthy erase times constrain test flexibility and may add to the total system cost.

Finally, PLDs are usually subjected to a complete, electrical QA test upon receipt, which typically adds 7 percent to the device cost. This additional cost is based on test engineering and manufacturing resources, yield and equipment utilization. Manufacturers can avoid this additional expense without degrading device quality, by using E²CMOS devices. These devices are 100 percent pretested by the manufacturer, and require no incoming test. And their instant erasability lets IC manufacturers perform extensive tests at the manufacturing stage, prior to shipment to end-users.

Some companies, however, have extensive incoming QA operations that can't be eliminated. Reusable E²CMOS devices are ideal for these operations because they can be returned to manufacturing inventory after QA testing, instantly reprogrammed, and reused in production boards. This flexibility also lets QA engineers perform their inspection at any step in the process.

QA engineers can also simplify their testing by using generic-architecture, macrocell-based devices. These devices can be tested with one common test program and then configure in many ways during the programming operation. This step eliminates the generation and maintenance of multiple test programs and fixtures, one for each fixed architecture.

Analyzing the system cost

Managers can reduce these overhead costs to a formula based on a simple approach that assumes a percentage cost adder and yield factor for each operation: $Cost_n = Cost_{n-1} + (Cost_{n-1}/Yield_n)$. This formula is generic, so managers can tailor the factors to their specific environment and then analyze the actual system cost of using a particular PLD.

E²CMOS PLDs offer performance, quality, reliability and, most important, cost advantages over alternative solutions. By eliminating yield and handling loss, improving quality, and simplifying inventory management, designers can significantly reduce the true cost of using PLDs.

Dean Suhr is product marketing manager at Lattice Semiconductor, Hillsboro, OR.

The Factor of Ten rule

It's crucial that managers keep in mind the cost of detecting and repairing defective PLDs during manufacturing — and the importance of early detection. A common guideline for determining this cost is the Factor of Ten rule.

This rule states that the cost of detecting and repairing a defective PLD grows by a factor of ten at each subsequent stage in the manufacturing process. This dramatic growth rate is possible because other symptoms mask the PLD's faulty functionality as the device is buried deeper in the system.

The Factor of Ten rule implies that the earlier defective devices are caught, the lower the repair cost. If defects aren't found early, a very small yield loss can be greatly magnified by the quantity of devices on a board or in a system. Even a yield loss as small as 0.5 percent can result in a 5 percent system failure rate with only five PLDs per system. A loss of 0.5 percent translates to a defect rate of 5,000 ppm, a high defect rate.

Section 1: Introduction to pLSI™ and ispLSI™

Section 2: pLSI and ispLSI Data Sheets

Section 3: GAL® Data Specifications

Section 4: pLSI and ispLSI Architecture

Section 5: pLSI and ispLSI Advantages

Section 6: pLSI and ispLSI Software Development Tools

Section 7: pLSI and ispLSI Design Optimization

Section 8: pLSI and ispLSI Application Notes

Section 9: pLSI and ispLSI Article Reprints

Section 10: pLSI and ispLSI Quality, Reliability & Programmer Support

Quality Assurance Program 10-1

Qualification Program 10-3

E²CMOS Testability Improves Quality 10-4

Package Thermal Resistance 10-5

Package Diagrams 10-6

pLSI and ispLSI 1032 Programmer Support 10-8

Section 11: Sales Information

Quality Assurance Program

Introduction

Lattice views quality assurance as a corporate responsibility and an integral part of all planning activities. Lattice's Quality Assurance organization is independent from Manufacturing and has direct access to top management, assuring that sufficient authority is afforded to quality issues.

Lattice's quality program is in full compliance with the quality assurance requirements of MIL-M-38510 Appendix A and all inspection system requirements of MIL-I-45208.

Qualification

All new products, processes and vendors must pass predefined evaluations before receiving initial qualification release. Major changes to products, processes or vendors require additional qualification before implementation.

In-Process Control

Qualified product must be manufactured under strict quality controls that start with regulated procurement and documented inspection plans for all incoming materials. Sample testing and in-line monitoring as well as statistical process control charts provide constant feedback at each critical step of the manufacturing process. Nonconforming material is identified, segregated, analyzed and dispositioned according to procedures which also require corrective action be specified to eliminate the cause of the defect. To assure continuing conformance to reliability requirements, an accelerated monitor program is maintained on all processes and wafer fabrication sites.

Calibration

All critical equipment involved in the manufacture, testing, or inspection of Lattice product must meet the requirements of our established calibration system which is in compliance to MIL-STD-45662.

Training

All Lattice manufacturing personnel complete a comprehensive training program and obtain formal certification for each production operation before they are allowed to manufacture products. Operators must be recertified on a periodic basis to assure ongoing compliance to all written procedures and specifications.

Subcontractor Control

All subcontracted operations must be performed by sources exhibiting a quality program commensurate to that of Lattice. These vendors are audited at least once each year to monitor their compliance to Lattice's Quality Assurance Program. Any major audit discrepancy requires corrective action and may result in disqualification.

Document Control

Lattice's document control system is under the direction of Quality Assurance and has the responsibility of assuring that every product has adequate written documentation released before production begins.

Drawings and specifications related to materials, processes, testing, products and subcontractors are maintained by the Document Control Department. A numbering system identifies each document by revision status, function and category.

Any change to existing documentation must be properly approved and released before implementation of the change. The change is implemented only if approved by the appropriate functional groups.

Control of Nonconforming Material

Identified failures from qualification testing, inspections, customer returns or in-process screening may be processed through Lattice's Failure Analysis group to determine the cause or relevancy of the failure and initiate corrective actions to eliminate the cause. All failure analysis reports are reviewed by Quality Assurance to convey awareness of any potential problems and assure that proper corrective action is taken.

Lattice has a Material Review Board (MRB) to investigate the cause of nonconformance and disposition the material. Lattice and customer specification requirements are thoroughly reviewed during MRB dispositions. The MRB consists of representatives from Manufacturing, Engineering, and Quality Assurance.

Finished Wafer Process Control Points for Commercial Devices

Step	Characteristics	Chart Type	Responsibility
Wafer Parametric Test	Test Structure Performance	Cp, K	Engineering
Die Functional Test	Functional AC, DC Performance	Trend	Engineering
Wafer Saw	Kerf Width	X-R	Production
Die Attach	Visual Defects	P	Production
Wire Bond	Pull Test	X-R	Quality Assurance
Mold	Mold Tool Temperature	X-R	Production
Deflash Trim Form	Visual Defects Coplanarity (PLCC only)	P X-R	Production Production
Solder Plate	Thickness % Pb	X-R P	Quality Assurance Quality Assurance
Assembly Final Visual	Visual Defects	P	Quality Assurance
Test	Functional AC, DC Performance	Trend	Engineering
Topside Mark	Visual Defects	P	Production
Final QA	Visual Defects	P	Quality Assurance
Final QA Test	Functional AC, DC Performance	P	Quality Assurance

Qualification Program

Introduction

Lattice has an intensive qualification program for examining and testing new products, processes, and vendors in order to ensure the highest levels of quality. Lattice's Reliability Engineering Group is responsible for defining and implementing this qualification program.

The following table outlines the steps which must be performed before a new product, package or process is qualified. The requirements listed below are general guidelines. Detailed information on Lattice's qualification process is available to customers upon request.

Qualification Requirements

Test	Number of Samples	Duration		
		New Product	New Wafer Process	New Package
125° C Operating Lifetest (5.25V)	300	1,000 Hours	2,000 Hours	2,000 Hours[1]
150° C Biased Retention Bake (5.25V)	450	1,000 Hours	2,000 Hours	2,000 Hours[1]
Endurance Cycling	75	10,000 Cycles	10,000 Cycles	N/A
ESD	48	End of Test	End of Test	N/A
Latch-up Immunity	27	End of Test	End of Test	N/A
Temperature Cycling (–65° to 150° C)	150	1,000 Cycles	1,000 Cycles	1,000 Cycles
Biased 85/85 (5V)	225	N/A	1,000 Hours	1,000 Hours
Autoclave (121° C, 15 psig)	150	N/A	336 Hours	336 Hours
Solderability	9	N/A	N/A	End of Test[1]
Bond Strength	12	N/A	N/A	End of Test[1]

1. Required for new assembly technologies only.

E²CMOS Testability Improves Quality

Introduction

The inherent testability of Lattice's E²CMOS PLDs significantly improves their quality and reliability. By using electrically erasable PROM technology to produce GAL devices, Lattice is able to perform 100% AC/DC, functional, and parametric testing of every single device. In order to achieve the highest quality levels, Lattice programs and tests each device repeatedly throughout the manufacturing process.

Actual Test vs. Simulated Test

Why is "actual test" so significant? PLDs, unlike most other semiconductor devices, have a programmable element that determines the final device functionality and AC/DC performance. These programmable elements can be fabricated from metal link fuses, programmable diodes or transistors, volatile static RAM cells, UV EPROM cells or electrically erasable EEPROM cells. Each of these technologies carries a different variability of programming success and a variance in the impact of the programming success on the performance and reliability of the device.

The most common programmable elements are the metal fuse, EPROM cell and EEPROM cell. Of these element types, only the EEPROM cell can be thoroughly tested by the manufacturer prior to shipment to an end user OEM.

EEPROM Allows Erasability and Actual Test

Each of the technologies identified above can be programmed. In this manner they are all the same. The differences become apparent when the erase times are analyzed. Metal link and One-Time Programmable (OTP) devices cannot be erased. UV EPROM devices can be erased, however they require a 20-30 minute erase time and an expensive windowed package. EEPROM devices, on the other hand, offer instant erasability on the order of 50 ms (thousandth's of a second). The advantage of this instant erase for manufacturing test is significant. Instant erase allows instant re-patterning for additional testing.

EEPROM technology has been used for PLD manufacturing by Lattice for more than half a decade. Lattice refers to their high performance EEPROM technology as E²CMOS technology. Extensive reliability studies of the technology have been performed with industry-wide acceptance, including the military.

Other Methods Are Imprecise

All PLD devices must be tested to some degree to validate functionality and performance. Technologies that are not erasable or offer lengthy erase times which severely constrain the test flexibility. Since the normal "user" programmable elements cannot be programmed during manufacture (all elements must be available for end-user programming) the manufacturers resort to using simulated and correlated performance of test rows, test columns and phantom or dummy-test arrays. At best, this is a statistical measure of the actual device performance. One need only look at the "normal" programming yield fallout of 0.5 —> 3% or the "acceptable" post-programming test vector & board yield fallout of 0.5 —> 2% to know that this correlation is weak. The quality systems of today are measuring defects in the parts per million (PPM). A six sigma program requires less than 3.4 PPM, four orders of magnitude less than that achievable with non-testable PLDs.

Actual Matrix Patterning

The unique capability of E²CMOS devices to be instantly electrically erased allows these devices to be patterned multiple times during Lattice's manufacturing test. Normal array cells in the programmable matrix are patterned, erased & tested again and again. The test rows or columns, phantom arrays, etc., that are used with other technologies are not necessary with E²CMOS devices. Programmability of every cell is checked dozens of times.

Actual Case AC/DC Testing

A PLD does not have a defined function until the engineer patterns the device with his custom pattern. The manufacturer, when considering the testing of a PLD, must consider the hundreds of different architecture and functional variations that can be created by the end user. Each configuration of architecture brings on a different set of actual pattern and stimulus conditions. Quick application of a series of worst case patterns that cover all of the permutations of input combinations, array load & switching, and output configuration is possible.

E²CMOS devices offer instant erasability to address this reconfiguration & test problem. Testing each additional configuration takes fractions of a second, allowing multiple patterns to be checked to assure performance to rated speeds. The final result is a device with defects reduced from PPH (Parts Per Hundred) to PPM (Parts Per Million).

Package Thermal Resistance

The following table provides information on the package thermal resistance of Lattice pLSI and ispLSI commercial grade devices.

Testing was performed per SEMI TEST METHOD G38-87: "Still and Forced-Air Junction to Ambient Thermal Resistance Measurements of IC Packages" with devices mounted on a thermal test board conforming to SEMI SPECIFICATION G42-88: Thermal Test Board Standardization for Measuring Junction-to-Ambient Thermal Resistance of Semiconductor Packages".

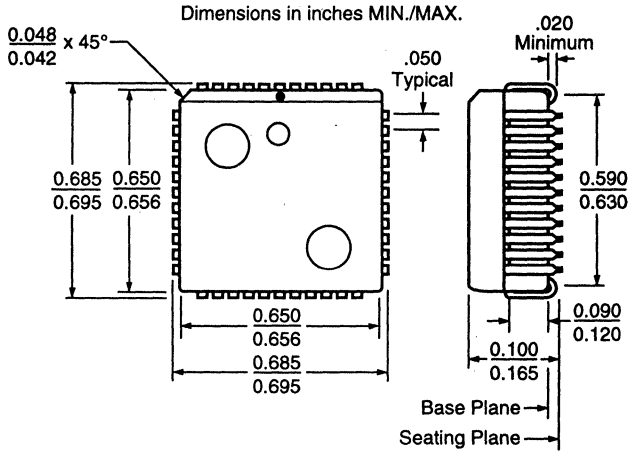
Package Thermal Resistance

Commercial Grade Devices

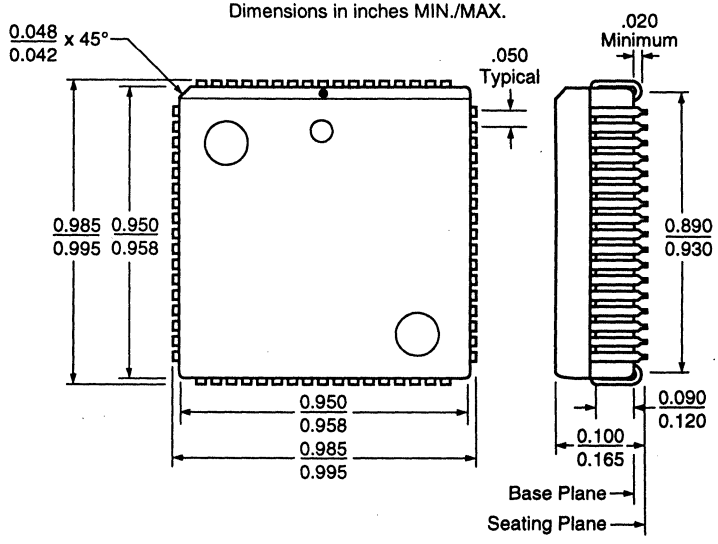
Package	Device Type	θ_{JA}	θ_{JC}
44-pin PLCC	pLSI 1016 ispLSI 1016	50° C/W	16° C/W
68-pin PLCC	pLSI 1024 ispLSI 1024	45° C/W	13° C/W
84-pin PLCC	pLSI 1032 ispLSI 1032	42° C/W	12° C/W
120-pin PQFP	pLSI 1048 ispLSI 1048	55° C/W	18° C/W

Package Diagrams

44-Pin PLCC

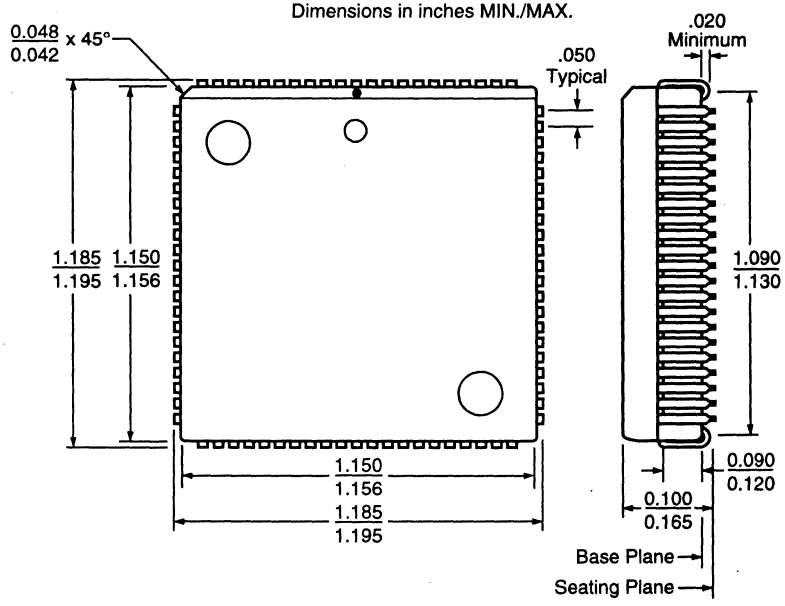


68-Pin PLCC



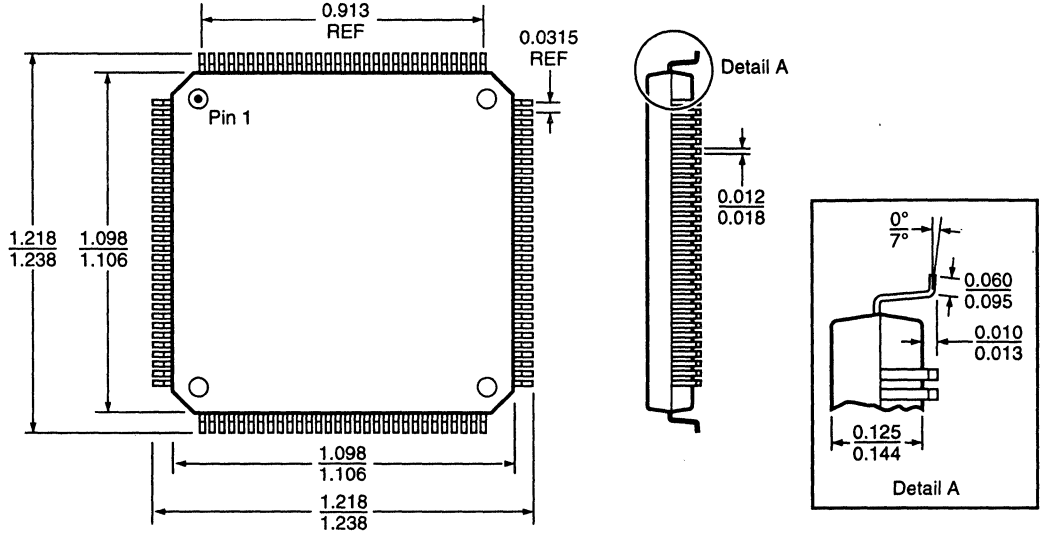
84-Pin PLCC

Dimensions in inches MIN./MAX.



120-Pin PQFP

Dimensions in inches MIN./MAX.



pLSI and ispLSI 1032 Programmer Support

Approved Programmer Vendors

Programmer Vendor	Programmer Model	Adapter	Revision Qualified	Test Vectors
Advin Systems	Pilot GL/U40	AM-LSI32A	10.24	No
	Pilot U84	AM-LSI32B	10.24	Yes
BP Microsystems	PLD1128	28-pin DIP to 84-pin PLCC[1]	1.82	No
Data I/O	2900	28-pin DIP to 84-pin PLCC[1]	V1.7	No
	3900	28-pin DIP to 84-pin PLCC[1]	V1.1	No
	3900	3900 PLCC	V1.2	Yes
	Unisite 40/48	28-pin DIP to 84-pin PLCC[1]	V3.6	No
	Unisite 48	PinSite	V3.6	Yes
Logical Devices	Allpro 32/40	28-pin DIP to 84-pin PLCC[1]	V2.1	No
	Allpro 88	No Adapter Required	V2.1	Yes
SMS Microsystems	Sprint Expert	28-pin DIP to 84-pin PLCC[1]	3.6	No
Stag	ZL30A	28-pin DIP to 84-pin PLCC[1]	V30B04	No
System General	Turpro 1	84-pin PLCC	1.43	No

1. Refer to Qualified 28-Pin DIP to 84-Pin PLCC Adapters table for adapter manufacturers.

Qualified 28-Pin DIP to 84-Pin PLCC Adapters

Manufacturer	Part Number	Address
EDI Corporation	84-PLCC/28DIP6-ZL-LSI1032	15507 Baldwin Road Patterson, CA 95363 Tel (209) 892-3270 FAX (209) 892-3610
Emulation Technology	842802P600-YAM	2344 Walsh Avenue, Building F Santa Clara, CA 95051 Tel (408) 982-0660 FAX (408) 982-0664

Programmer Vendors

Advin Systems
1050-L Duane Ave
Sunnyvale, CA 94086
Phone: (408) 243-7000
FAX: (408) 736-2503

BP Microsystems
10681 Haddington
Suite #190
Houston, TX 77043
Tel: (713) 461-9430
FAX: (713) 461-7431
BBS: (713) 461-4958

Data I/O Corp.
10525 Willows Road N.E.
P.O. Box 97046
Redmond, WA 98073-9746
Phone: (206) 881-6444
FAX: (206) 882-1043
In Europe contact:
Data I/O Corp.
Phone +31 (0) 20-6622866
In Japan contact:
Data I/O Corp.
Phone: (03) 432-6991

Logical Devices
1321 N.W. 65th Place
Fort Lauderdale, FL 33309
Phone: (305) 974-0967
FAX: (305) 974-8531

SMS Micro Systems
1M Morgenthal
D-8994 Hergatz
Scwarzenberg
West Germany
In the U.S. contact:
Encore Technology Corp.
13720 Midway Suite 105
Dallas, TX 75244
Tel: (214) 233-2614
FAX: (214) 233-3122

Stag Microsystems
Martinfeld
Welwyn Garden City
Hertz. AL7 15T
United Kingdom
Phone: 011-44-707-332148
FAX: 011-44-707-371503
In the U.S. contact:
Stag Microsystems
1600 Wyatt Dr.
Santa Clara, CA 95054
Phone: (408) 988-1118
FAX: (408) 988-1232

System General
3Fl., No. 6, Lane 4
Tun Hwa N. Rd.
P.O. Box: 53-591
Taipei, Taiwan R.O.C.
Phone: 886-2-7212613
FAX: 866-2-7212615
In the U.S. contact:
System General
244 S. Hillview Dr.
Milpitas, CA 95035
Phone: (408) 263-6667
FAX: (408) 262-9220

Section 1: Introduction to pLSI™ and ispLSI™

Section 2: pLSI and ispLSI Data Sheets

Section 3: GAL® Data Specifications

Section 4: pLSI and ispLSI Architecture

Section 5: pLSI and ispLSI Advantages

Section 6: pLSI and ispLSI Software Development Tools

Section 7: pLSI and ispLSI Design Optimization

Section 8: pLSI and ispLSI Application Notes

Section 9: pLSI and ispLSI Article Reprints

Section 10: pLSI and ispLSI Quality, Reliability & Programmer Support

Section 11: Sales Information

Sales Offices 11-1

11

Sales Offices

LATTICE SALES OFFICES

FRANCE
Lattice Semiconductor
Les Bureaux de Sèvres
72-78, Grand Rue
92310 Sèvres
TEL: (33) 1 45 34 10 10
FAX: (33) 1 46 26 71 36

GERMANY
Lattice Semiconductor
Stahlgruberring 12
8000 Munich 82
TEL: (49) 89 42 01 107
FAX: (49) 89 422 731

JAPAN
Lattice Semiconductor
Peony Kikuchi 201
1-8-4, Botan
Koto-ku, Tokyo 135
TEL: 33-642-0621
FAX: 33-642-0629

UNITED KINGDOM
Lattice Semiconductor
Lords Court
St. Leonard's Road
Windsor
Berkshire SL4 3DB
TEL: 753-830-842
FAX: 753-833-457

NORTH AMERICA

CALIFORNIA
Lattice Semiconductor
4000 Burton Drive
Santa Clara, CA 95051
TEL: (408) 980-7878
FAX: (408) 980-0839

Lattice Semiconductor
Carlsbad Pacific Ctr. One
701 Palomar Airport Rd.
3rd Floor
Carlsbad, CA 92009
TEL: (619) 931-4751
FAX: (619) 431-1821

GEORGIA
Lattice Semiconductor
3105 Medlock Bridge Rd.
Norcross, GA 30071
TEL: (404) 446-2930
FAX: (404) 416-7404

MASSACHUSETTS
Lattice Semiconductor
67 S. Bedford St.
Suite 400 West
Burlington, MA 01803
TEL: (617) 229-5819
FAX: (617) 272-3213

MINNESOTA
Lattice Semiconductor
TEL: (612) 891-5200
FAX: (612) 891-5205

NEW JERSEY
Lattice Semiconductor
TEL: (201) 744-5908
FAX: (201) 509-9309

OREGON
Lattice Semiconductor
5555 N.E. Moore Ct.
Hillsboro, OR 97124
TEL: (503) 780-6771
FAX: (503) 681-3037

TEXAS
Lattice Semiconductor
100 Decker Ct. Ste. 280
Irving, TX 75062
TEL: (214) 650-1236
FAX: (214) 650-1237

NORTH AMERICAN SALES REPRESENTATIVES

ALABAMA
CSR Electronics, Inc.
303 Williams Avenue
Ste. 931
Huntsville, AL 35801
TEL: (205) 533-2444
FAX: (205) 536-4031

ARIZONA
Summit Sales
7802 E. Gray Rd. #600
Scottsdale, AZ 85260
TEL: (602) 998-4850
FAX: (602) 998-5274

ARKANSAS
West Associates
Richardson, TX 75081
TEL: (214) 680-2800
FAX: (214) 699-0330

CALIFORNIA
Bager Electronics
17220 Newhope St. #209
Fountain Valley, CA 92708
TEL: (714) 957-3367
FAX: (714) 546-2654

Bager Electronics
6324 Variel Ave. #314
Woodland Hills, CA 91367
TEL: (818) 712-0011
FAX: (818) 712-0160

Criterion Sales
3350 Scott Blvd, Bldg.44
Santa Clara, CA 95054
TEL: (408) 988-6300
FAX: (408) 986-9039

Earle Associates
7585 Ronson Rd. #200
San Diego, CA 92111
TEL: (619) 278-5441
FAX: (619) 278-5443

COLORADO
Waugaman Associates
4800 Van Gordon
Wheat Ridge, CO 80033
TEL: (303) 423-1020
FAX: (303) 467-3095

CONNECTICUT
Comp Rep Associates
P.O Box 311
21 Hillside Avenue
West Haven, CT 06516
TEL: (203) 269-1145
FAX: (203) 269-2819

FLORIDA
Sales Engineering Concepts
776 S. Military Trail
Deerfield Beach, FL 33442
TEL: (305) 426-4601
FAX: (305) 427-7338

Sales Engineering Concepts
600 S. Norhtlake Blvd. #230
Altamonte Spgs, FL 32701
TEL: (407) 830-8444
FAX: (407) 830-8684

GEORGIA
CSR Electronics, Inc.
1651 Mt. Vernon Rd. # 200
Atlanta, GA 30338
TEL: (404) 396-3720
FAX: (404) 394-8387

ILLINOIS
Heartland Tech Mktg.
1615 Colonial Pkwy.
Inverness, IL 60067
TEL: (708) 358-6622
FAX: (708) 358-7660

IOWA
Stan Clothier Company
1930 St. Andrews, N.E.
Cedar Rapids, IA 52402
TEL: (319) 393-1576
FAX: (319) 393-7317

KANSAS
Stan Clothier Company
605 Clairborne, Ste. 275C
Olathe, Kansas 66062
TEL: (913) 829-0073
FAX: (913) 829-0429

LOUISIANA
West Associates
Richardson, TX 75081
TEL: (214) 680-2800
FAX: (214) 699-0330

MARYLAND
Deltatronics
24048 Sugar Cane Ln.
Gaithersburg, MD 20882
TEL: (301) 253-0615
FAX: (301) 253-9108

MASSACHUSETTS
Comp Rep Associates
100 Everett Street
Westwood, MA 02090
TEL: (617) 329-3454
FAX: (617) 329-6395

MICHIGAN
Greiner & Associates
15324 E. Jefferson Ave.
Grosse Pointe Park, MI 48230
TEL: (313) 499-0188
FAX: (313) 499-0665

MINNESOTA
Stan Clothier Company
9600 W. 76th St., Ste. #A
Eden Prairie, MN 55344
TEL: (612) 944-3456
FAX: (612) 944-6904

Sales Offices

MISSOURI
Stan Clothier Company
3910 Old Highway 94 South
Suite 116
St. Charles, MO 63304
TEL: (314) 928-8078
FAX: (314) 447-5214

MONTANA
Waugaman Associates
Wheat Ridge, CO 80033
TEL: (303) 423-1020
FAX: (303) 467-3095

NEW HAMPSHIRE
Comp Rep Associates
Westwood, MA 02090
TEL: (617) 329-3454
FAX: (617) 329-6395

NEW JERSEY
Technical Marketing Group
175-3C Fairfield Rd.
West Caldwell, NJ 07006
TEL: (201) 226-3300
FAX: (201) 226-9518

NEW MEXICO
Summit Sales
Scottsdale, AZ
TEL: (602) 998-4850
FAX: (602) 998-5274

NEW YORK
Technical Marketing Group
20 Broad Hollow Rd.
Melville, NY 11747
TEL: (516) 351-8833
FAX: (516) 351-8667

Tri-Tech Electronics
300 Main St.
E. Rochester, NY 14445
TEL: (716) 385-6500
FAX: (716) 385-7655

Tri-Tech Electronics
14 Westview Dr.
Fishkill, NY 12524
TEL: (914) 897-5611
FAX: (914) 897-5611

Tri-Tech Electronics
6836 E. Genesee St.
Fayetteville, NY 13066
TEL: (315) 446-2881
FAX: (315) 446-3047

NORTH CAROLINA
CSR Electronics, Inc.
5848 Faringdon Place, Ste. 2
Raleigh, NC 27609
TEL: (919) 878-9200
FAX: (919) 878-9117

CSR Electronics, Inc.
8020 Crockett Court
Charlotte, NC 28226
TEL: (704) 847-5806
FAX: (704) 847-0056

OHIO
Makin & Associates
3165 Lynwood Rd.
Cincinnati, OH 45208
TEL: (513) 871-2424
FAX: (513) 871-2524

Makin & Associates
6400 Riverside Dr. Bldg. A
Dublin, OH 43017
TEL: (614) 793-9545
FAX: (614) 793-0256

Makin & Associates
32915 Aurora Ave. #270
Solon, OH 44139
TEL: (216) 248-7370
FAX: (216) 248-7372

OKLAHOMA
West Associates
9717 E. 42nd St. #125
Tulsa, OK 74146
TEL: (918) 665-3465
FAX: (918) 663-1762

OREGON
Northwest Marketing
6975 SW Sandburg Rd. #330
Portland, OR 97223
TEL: (503) 620-0441
FAX: (503) 684-2541

PENNSYLVANIA
Deltatronics
921 Penllyn Pike
Blue Bell, PA 19422
TEL: (215) 641-9930
FAX: (215) 641-9934

SOUTH CAROLINA
CSR Electronics
Raleigh, N.C. 27609
TEL: (919) 878-9200
FAX: (919) 878-9117

TENNESSEE
CSR Electronics, Inc.
Andrew Johnson Office Pl.
912 S. Gay Street, Ste. 1218
Knoxville, TN 37902
TEL: (615) 637-0293
FAX: (615) 637-0466

TEXAS
West Associates
4615 Southwest Fwy #720
Houston, TX 77027
TEL: (713) 621-5983
FAX: (713) 621-5895

West Associates
9171 Capital of Texas
North Houston Bldg. #120
Austin, TX 78759
TEL: (512) 343-1199
FAX: (512) 343-1922

West Associates
801 E. Campbell Rd. #350
Richardson, TX 75081
TEL: (214) 680-2800
FAX: (214) 699-0330

UTAH
Waugaman Associates
876 East Vine St.
Murray, UT 84107
TEL: (801) 261-0802
FAX: (801) 261-0830

VIRGINIA
Deltatronics
Blue Bell, PA 19422
TEL: (215) 641-9930
FAX: (215) 641-9934

WASHINGTON
Northwest Marketing
12835 Bel-Red Rd. #330N
Bellevue, WA 98005
TEL: (206) 455-5846
FAX: (206) 451-1130

WASHINGTON D.C.
Deltatronics
Gaithersburg, MD 20882
TEL: (301) 253-0615
FAX: (301) 253-9108

WISCONSIN
Heartland Technical Mktg.
18000 W. Sarah Lane
Suite 340
Brookfield, WI 53045
TEL: (414) 792-0920
FAX: (414) 792 0929

PUERTO RICO
Sales Engineering Concepts
URB Mercedita
Condo. Buena Vista C-1
Ponce, P.R. 00731
TEL: (809) 841-4220
FEX: (809) 259-7223

CANADA

ALBERTA
Dynasty Components
Calgary, Alberta
TEL: (403) 560-1212
FAX: (613) 723-8820

BRITISH COLUMBIA
Dynasty Components
Vancouver, British Columbia
TEL: (604) 597-0068
FAX: (613) 723-8820

ONTARIO
Dynasty Components
174 Colonade Rd. S.
Unit 21
Nepean, Ontario
Canada, K2E 7J5
TEL: (613) 723-0671
FAX: (613) 723-8820

Dynasty Components
Toronto, Ontario
TEL: (416) 672-5977
FAX: (613) 723-8820

QUEBEC
Dynasty Components
Montreal, Quebec
TEL: (514) 843-1879
FAX: (613) 723-8820

INTERNATIONAL SALES REPRESENTATIVES AND DISTRIBUTORS

AUSTRALIA
RAE Industrial Elect. Pty.
P.O Box 76
Thirroul, NSW 2515
TEL: (61) 42-673722
FAX: (61) 42-681075

ZATEK Components, Ltd.
1059 Victoria Road
P.O Box 397, Suite 8
West Ryde, NSW 2114
Sidney, 3153
TEL: (61) 2 874-0122
FAX: (61) 2 874-6171

ZATEK Components, Ltd.
Level 2, 96 Phillip Street
Parramatta 2150
P.O Box 1077
Parramatta 2150
TEL: (61) 2 895-5534
FAX: (61) 2 895-5535

AUSTRIA
Ing. E. Steiner GmbH.
Hummelgasse 14
A-1130 Wien
TEL: (43) 222-827-4740
FAX: (43) 222-828-5617

BELGIUM
Alcom Electronics B.V.B.A.
Singel 3
2550 Kontich
TEL: (32) 3 458-3033
FAX: (32) 3 458 3126

DENMARK
Ditz Schweitzer
Vallensbaekvej 41
Postboks 5,
DK-2605 Brendby
TEL: (45) 42 45 30 44
FAX: (45) 42 45 92 06
TLX: 85533257

FINLAND
Telercas OY
Luomannotko 6
02200 Espoo
TEL: (358) 804 521622
FAX: (358) 804 523337
TLX: 857123212

FRANCE
Franelec
50 Rue de L'Aubepine
B.P. 158
92185 Antony Cedex
TEL: (33) 1 4096 0909
FAX: (33) 1 4096 0250
TLX: 842250067

DataDis
3 Bis Rue Rene Cassin
B.P 84
91303 Massey Cedex
TEL: (33) 69-20 4141
FAX: (33) 69-20 4900

GERMANY
Alfatron GmbH.
Stahlgruberring 12
8000 Munich 82
TEL: (49) 89 4204 910
FAX: (49) 89 4204 9159
TLX: (49) 5216935

EBV/MHV
Hans-Pinsel-Str.4
8013 Haar bei Munchen
TEL: (49) 89 420-1107
FAX: (49) 89 468-808

HONG KONG
RTI Industries Co. Ltd.
A19, 10th Floor
Proficient Ind. Centre
6, Wang Kwan Rd.
Kowloon
TEL: (852) 795 7421
FAX: (852) 795 7839

INDIA
Hindetron
33/44A, 8th Main Road
Rajmahal Vilas Ext.
Bangaore 560-080
TEL: (91) 812 348 266
FAX: (91) 812 345 022
TLX: (91) 08452741HSPL IN

IRELAND
Silicon Concepts
3 Mills View Close
Dukesmeadow, Kilkenny
Kilkenny County
TEL: (353) 566 4002

ISRAEL
Unitec
Rechov Maskit 1
Herzlia B, PO Box 2123
46120
TEL: (972) 52 576006
FAX: (972) 52 576790
TLX: 922341990

ITALY
Comprel S.R.L.
Viale F. Testi, 115
20092 Cinosello B.
Milano
TEL: (39) 2-61206415
FAX: (39) 2-6128158

Comprel S.P.A.
Viale F. Testi, 115
20092 Cinosello B.
Milano
TEL: (39) 2-6120641
FAX: (39) 2-61280526

JAPAN
Ado Electronic Indust. Co.
4th Floor, Fukui Building
No. 2-12 Sotokanda
2-Chome, Chiyoda-ku
Tokyo 101
TEL: (81) 3-3257-2600
FAX: (81) 3-3251-6796
TLX: 7812224754

Japan Macnics Corp.
Hakusan High-Tech Park
1-22-2 Hakusan-cho, Midori-ku
Yokohama, 226
TEL: (81) 45-939-6140
FAX: (81) 45-939-6141
TLX: 78128988

Hakuto Company, Ltd.
2-29, Toranomon, 1 chome
Minato-Ku, Tokyo 105
TEL: (81) 3-3597-8910
FAX: (81) 3-3597-8975
TLX: J22912BRAPAN

Hoei Denki
6-60, 2-Chome, Niitaka
Chiyoda-Ku,
Osaka 532
TEL: (81) 6 394 1113
FAX: (81) 6 396 5647
TLX: 5233694HOEIDK J

KOREA
Ellen & Company
Suite #302 Ilbok Bldg.
1602-4 Seocho-Dong
Seocho-ku, Seoul
TEL: (82) 02 587 5724
FAX: (82) 02 585 1519

NETHERLANDS
Alcom Electronics B.V.
Essebaan 1
2908 LJ Capelle Aan
Den Ijssel
TEL: (31) 10 4519533
FAX: (31) 10 4586482
TLX: 26160

NORWAY
Henaco A/S
Trondheimsveien 436
Ammerud
Oslo 9
TEL: (47) 2162110
FAX: (47) 2257780
TLX: 76716

SINGAPORE
Technology Distribution
14 Sungei Kadut Ave. #03-00
Singapore 2572
TEL: (65) 368 6065
FAX: (65) 368 0182

SOUTH AFRICA
Multikomponent
Cnr. Vanacht & Gewel St.
Isando 1600, P.O Box 695
TEL: (27) 11 974 1525
FAX: (27) 11 392 2463
TLX: 960426905

SPAIN
ATD
Avenida de la Industria
No. 32, 2B
28100 Alcobendas
Madrid
TEL: (34) 1 661-6551
FAX: (34) 1 661 6300

SWEDEN
Pelcon Electronics
Fagerstagatan 6-8
S-163 08 Spanga
TEL: (46) 8 795 9870
FAX: (46) 8 760 7685

SWITZERLAND
Ascom Primotec AG
Täfernstrasse 37
CH-5405 Baden-Dättwil
TEL: (41) 5684-0171
FAX: (41) 5683-3454
TLX: 828 221 apri ch

TAIWAN
Master Electronics
16F, No. 810
Tunghua South Road
Taipei, Taiwan
TEL: (886) 2-735-7316
FAX: (886) 2-735-0902

UNITED KINGDOM
Macro Marketing
Burnham Lane
Slough SL1 6LN
England
TEL: (44) 628 604383
FAX: (44) 628 666873
TLX: 851847945

Micro Call
17 Thame Park Rd.
Thame, Oxon 0X9 3XD
England
TEL: (44) 84 426-1939
FAX: (44) 84 426-1678

Sales Offices

Silicon Concepts, Ltd.
PEC Lynchborough Rd.
Passfield, Liphook
Hampshire GU30 7SB
England
TEL: (44) 428 751617
FAX: (44) 428 751603

Silicon Concepts, Ltd.
Meridale, Welsh Street
Chepston, Gwent, NP6 5LR
Wales
TEL: (44) 291-624101
FAX: (44) 291-629878

NORTH AMERICAN DISTRIBUTORS

ALABAMA
Arrow Electronics
1015 Henderson Rd.
Huntsville, AL 35816
(205) 837-6955

Hall-Mark Electronics
4890 University Square
Suite 1
Huntsville, AL 35816
(205) 837-8700

Marshall Industries
3313 Memorial Pkwy S.
Huntsville, AL 35801
(205) 881-9235

ARIZONA
Arrow Electronics
2415 W. Erie Drive
Tempe, AZ 85282
(602) 431-0030

Hall-Mark Electronics
4637 S. 36th Place
Phoenix, AZ 85040
(602) 437-1200

Insight Electronics
1525 W. University Dr.
Suite #105
Tempe, AZ 85281
(602) 829-1800

Marshall Industries
9830 S. 51st St. #B121
Phoenix, AZ 85044
(602) 496-0290

NORTHERN CALIFORNIA
Arrow Electronics
1180 Murphy Ave.
San Jose, CA 95131
(408) 441-9700

Hall-Mark Electronics
580 Menio Drive
Suite 2
Rocklin, CA 95677
(916) 624-9781

Hall-Mark Electronics
2105 Lundy Avenue
San Jose, CA 95131
(408) 432-4000

Insight Electronics
1295 Oakmead Pkwy.
Sunnyvale, CA 94086
(408)720-9222

Marshall Industries
336 Los Coches St.
Milpitas, CA 95035
(408) 942-4600

Marshall Industries
3039 Kilgore Ave. #140
Rancho Cordova, CA 95670
(916) 635-9700

SOUTHERN CALIFORNIA
Arrow Electronics
Malibu Canyon Bus. Park
26707 W. Agoura Road
Calabasas, CA 91302
(818) 880-9686

Arrow Electronics
6 Cromwell, Suite 100
Irvine, CA 92718
(714) 587-0404

Arrow Electronics
9511 Ridgehaven Ct.
San Diego, CA 92123
(619) 565-4800

Hall-Mark Electronics
3878 B-Ruffin Road
San Diego, CA 92123
(619) 268-1201

Hall-Mark Electronics
1 Mauchly
Irvine, CA 92718
(714) 727-6000

Hall-Mark Electronics
9420 Topanga Canyon
Chatsworth, CA 91311
(818) 773-4500

Insight Electronics
28038 Dorothy Dr. #2
Aquora, CA 91301
(818) 707- 2100

Insight Electronics
6885 Flanders Dr. #C
San Diego, CA 92121
(619) 587-0471

Insight Electronics
15635 Alton Pkwy. #120
Irvine, CA 92718
(714) 727-3291

Marshall Industries
26637 Agoura Rd.
Calabasas, CA 91302
(818) 878-7000

Marshall Industries
9320 Telstar Ave.
El Monte, CA 91731-3004
(818) 307-6000

Marshall Industries
One Morgan
Irvine, CA 92718
(714) 458-5301

Marshall Industries
10105 Carroll Canyon Rd.
San Diego, CA 92131
(619) 578-9600

COLORADO
Arrow Electronics
61 Inverness Drive East
Suite 105
Englewood, CO 80112
(303) 373-5616

Hall-Mark Electronics
12503 E. Euclid Drive
Suite 20
Englewood, CO 80111
(303) 790-1662

Marshall Industries
12351 N. Grant
Thornton, CO 80241
(303) 451-8383

CONNECTICUT
Arrow Electronics
12 Beaumont Rd.
Wallingford, CT 06492
(203) 265-7741

Hall-Mark Electronics
615 West Johnson Ave.
Bldg. 3
Chesire, CT 06410
(203) 271-2844

Marshall Industries
20 Sterling Dr.
PO Box 200
Wallingford, CT 06492
(203) 265-3822

FLORIDA
Arrow Electronics
400 Fairway Dr.
Deerfield Beach, FL 33441
(305) 429-8200

Arrow Electronics
37 Skyline Dr.
Bldg. D, Suite 3101
Lake Mary, FL 32746
(407) 333-9300

Hall-Mark Electronics
10491 72nd North
Largo, FL 34637
(813) 541-7440

Hall-Mark Electronics
3161 SW 15th Street
(McNabb Road)
Pompano Beach, FL 33069
(305) 971-9280

Hall-Mark Electronics
489 E. Semoran Blvd.
Suite 145
Casselberry, FL 32707
(407) 830-5855

Marshall Industries
380 S. Northlake Rd. #1024
Altamonte Springs, FL 32701
(407) 767-8585

Marshall Industries
2700 Cypress Ck. Rd. #D114
Ft. Lauderdale, FL 33309
(305) 977-4880

Marshall Industries
2840 Scherer Dr. #410
St. Petersburg, FL 33716
(813) 573-1399

GEORGIA
Arrow Electronics
4205E River Green Pkwy.
Duluth, GA 30136
(404) 497-1300

Hall-Mark Electronics
3425 Corporate Way
Suite A
Duluth, GA 30136-2552
(404) 623-4400

Marshall Industries
5300 Oakbrook Pkwy #140
Norcross, GA 30093
(404) 923-5750

IOWA
Arrow Electronics
375 Collins Rd. NE
Cedar Rapids, IA 52402
(319) 395-7230

ILLINOIS
Arrow Electronics
1140 W. Thorndale Ave.
Itasca, IL 60143
(312) 250-0500

Hall-Mark Electronics
210 Mittel Drive
Wood Dale, IL 60191
(708) 860-3800

Marshall Industries
50 E. Commerce Dr. # 1
Schaumberg, IL 60173
(708) 490-0155

INDIANA
Arrow Electronics
7108 Lakeview Pkwy. W.
Indianapolis, IN 46268
(317) 299-2071

Hall-Mark Electronics
4275 West 96th Street
Indianapolis, IN 46268
(317) 872-8875

Marshall Industries
6990 Corporate Dr.
Indianapolis, IN 46278
(317) 297-0483

KANSAS
Arrow Electronics
9108 Legler Road
Lenexa, KS 66214
(913) 541-9542

Hall-Mark Electronics
10809 Lakeview Drive
Lenexa, KS 66215
(913) 888-4747

Marshall Industries
10413 W. 84th Ter.
Pine Ridge Business Park
Lenexa, KS 66214
(913) 492-3121

MARYLAND
Arrow Electronics
9800 J Patuxent Wood Dr.
Columbia, MD 21046
(301) 596-7800

Hall-Mark Electronics
10240 Old Columbia Rd.
Columbia, MD 21046
(301) 988-9800

Marshall Industries
2221 Broadbirch Dr.
Silver Springs, MD 20904
(301) 622-1118

MASSACHUSETTS
Arrow Electronics
25 Upton Dr.
Wilmington, MA 01887
(508) 658-0900

Hall-Mark Electronics
Pinehurst Park
6 Cook Street
Billerica, MA 01821
(508) 667-0902

Marshall Industries
33 Upton Dr.
Wilmington, MA 01887
(508) 658-0810

MICHIGAN
Arrow Electronics
3510 Roger Chaffee Mem. Blvd.
Grand Rapids, MI
(616) 243-0914

Arrow Electronics
19880 Haggerty Rd.
Livonia, MI 48152
(313) 462-2290

Hall-Mark Electronics
38027 Schoolcraft Road
Livonia, MI 48150
(313) 462-1205

Marshall Industries
31067 Schoolcraft
Livonia, MI 48150
(313) 525-5850

MINNESOTA
Arrow Electronics
10100 Viking Drive # 100
Eden Prairie, MN 55344
(612) 941-5280

Hall-Mark Electronics
10300 Valley View Road
Suite 101
Eden Prairie, MN 55344
(612) 941-2600

Marshall Industries
3955 Annapolis Lane
Plymouth, MN 55447
(612) 559-2211

MISSOURI
Arrow Electronics
2380 Schuetz Rd.
St. Louis, MO 63146
(314) 567-6888

Hall-Mark Electronics
4873 Rider Trail South
Earth City, MO 63045
(314) 291-5350

Marshall Industries
3377 Hollenberg Dr.
Bridgeton, MO 63044
(314) 291-4650

NEW JERSEY
Arrow Electronics
4 East Stow Rd. Unit 11
Marlton,NJ 08053
(609) 596-8000

Arrow Electronics
43 Route 46 East
Pinebrook, NJ 07058
(201) 227-7880

Hall-Mark Electronics
225 Executive Drive
Suite 5
Moorestown, NJ 08057
(609) 235-1900

Hall-Mark Electronics
200 Landidex Plaza
2nd Floor
Parsippany, NJ 07054
(201) 515-3000

Marshall Industries
101 Fairfield Rd.
Fairfield, NJ 07006
(201) 882-0320

Marshall Industries
158 Gaither Dr.
Mt. Laurel, NJ 08054
(609) 234-9100

NEW YORK
Arrow Electronics
25 Hub Drive
Melville, NY 11747
(516) 391-1300

Arrow Electronics
20 Oser Ave.
Hauppauge, NY 11788
(516) 231- 1000

Arrow Electronics
3375 Brighton-Henrietta
Townline Rd.
Rochester, NY 14623
(716) 427-0300

Hall-Mark Electronics
6605 Pittsford-Palmyra
Suite E8
Fairport, NY 14450
(716) 425-3300

Hall-Mark Electronics
3075 Veterans Memorial
Ronkonkoma, NY 11779
(516) 737-0600

Marshall Industries
275 Oser Ave.
Hauppauge, NY 11788
(516) 273-2424

Marshall Industries
1250 Scottsville Rd.
Rochester, NY 14624
(716) 235-7620

Marshall Industries
100 Marshall Drive
Endicott, NY 13790
(607) 785-2345

Sales Offices

NORTH CAROLINA
Arrow Electronics
5240 Greens Dairy Rd.
Raleigh, NC 27604
(919) 876-3132

Hall-Mark Electronics
5234 Greens Dairy Road
Raleigh, NC 27604
(919) 872-0712

Marshall Industries
5224 Greens Dairy Rd.
Raleigh, NC 27604
(919) 878-9882

OHIO
Arrow Electronics
6573E Cochran Rd.
Solon, OH 44139
(216) 248-3990

Arrow Electronics
8200 Washington Village Dr. #A
Centerville, OH 45458
(513) 435-5563

Hall-Mark Electronics
777 Dearborne Park Lane
Worthington, OH 43085
(614) 888-3313

Hall-Mark Electronics
5821 Harper Road
Solon, OH 44139
(216) 349-4632

Marshall Industries
3520 Park Center Dr.
Dayton, OH 45414
(513) 898-4480

Marshall Industries
30700 Bainbridge Rd. Unit A
Solon, OH 44139
(216) 248-1788

OKLAHOMA
Arrow Electronics
12111 East 51st St. #101
Tulsa, OK 74146
(918) 252-7537

Hall-Mark Electronics
5411 S. 125th East Ave.
Tulsa, OK 74146
(918) 254-6110

OREGON
Arrow Electronics
1885 N.W. 169th Place
Beaverton, OR 97006
(503) 629-8090

Insight Electronics
8705 SW Nimbus #200
Tigard, OR 97005
(503) 644-3300

Marshall Industries
9705 SW Gemini Dr.
Beaverton, OR 97005
(503) 644-5050

PENNSYLVANIA
Marshall Industries
401 Parkway View Dr.
Pittsburgh, PA 15205
(412) 788-0441

TEXAS
Arrow Electronics
2227 West Braker Lane
Austin, TX 78758
(512) 835-4180

Arrow Electronics
3220 Commander Dr.
Carrollton, TX 75006
(214) 380-6464

Arrow Electronics
10899 Kinghurst Dr. #100
Houston, TX 77099
(713) 530-4700

Hall-Mark Electronics
11420 Pagemill Road
Dallas, TX 75243
(214) 553-4300

Hall-Mark Electronics
Corporate Headquarters
11333 Pagemill Road
Dallas, TX 75243
(214) 343-5000

Hall-Mark Electronics
12211 Technology Blvd.
Austin, TX 78727
(512) 258-8848

Hall-Mark Electronics
8000 Westgien
Houston, TX 77063
(713) 781-6100

Insight Electronics
12703-A Research Blvd. #1
Austin, TX 78759
(512) 467-0800

Insight Electronics
1778 Plano Rd. #320
Richardson, TX 75081
(214) 783-0800

Insight Electronics
15437 McKaskle
Sugarland, TX 77478
(713) 448-0800

Marshall Industries
8504 Cross Park Dr.
Austin, TX 78754
(512) 837-1991

Marshall Industries
7250 Langtry
Houston, TX 77040
(713) 895-9200

Marshall Industries
2045 Chenault Street
Carrollton, TX 75006
(214) 233-5200

UTAH
Arrow Electronics
1946 West Parkway Blvd.
Salt Lake City, UT 84119
(801) 973-6913

Marshall Industries
2355 South 1070 West
Salt Lake City, UT 84119
(801) 973-2288

WASHINGTON
Arrow Electronics
14360 S.E. Eastgate Way
Bellevue, WA 98007
(206) 643-9992

Hall-Mark Electronics
250 NW 39th
Suite 4
Seattle, WA 98107

Insight Electronics
12002 115th Avenue, NE
Kirkland, WA 98034
(206) 820-8100

Marshall Industries
11715 N. Creek Pkwy. S.
Suite 112
Bothell, WA 98011
(206) 486-5747

WISCONSIN
Arrow Electronics
200 North Patrick Blvd.
Brookfield, WI 53005
(414) 792-0150

Hall-Mark Electronics
16255 West Lincoln Ave.
New Berlin, WI 53151
(414) 797-7844

Marshall Industries
20900 Swenson Dr. #150
Waukesha, WI 53186
(414) 797-8400

CANADA

ALBERTA
Future Electronics
3833-29th Street
Calgary, Alberta
Canada, T2A 5N1
(403) 250-5550

Future Electronics
4606-97th Street
Edmonton, Alberta
Canada, T6E 5N9
(403) 438-2858

BRITISH COLUMBIA
Arrow Electronics
8544 Baxter Place
Burnaby, British Columbia
Canada, V5A 4T8

Future Electronics
1695 Boundary Road
Vancouver, British Columbia
Canada, V5K 4X7
(604) 294-1166

MANITOBA
Future Electronics
100 King Edward
Winnipeg, Manitoba
Canada, R3H 0N8
(204) 786-7711

ONTARIO
Arrow Electronics
36 Antares Dr. Unit 100
Nepean, Ontario
Canada, K2E 7W5
(613) 226-6903

Arrow Electronics
1093 Meyerside Dr.
Mississauga, Ontario
Canada, L5P 1M4
(416) 670-7769

Future Electronics
1050 Baxter Road
Ottawa, Ontario
Canada, K2C 3P2
(613) 820-8313

Future Electronics
5935 Airport Rd., #200
Mississauga, Ontario
Canada, L4V 1W5
(416) 612-9200

Marshall Industries
4 Paget Rd.
Bldg. 1112, Unit 10
Brampton, Ontario
Canada, L6T 5G3
(416) 458-8046

QUEBEC
Arrow Electronics
1100 St. Regis Blvd.
Dorval, Quebec
Canada, H9P 2T5
(514) 421-7411

Future Electronics
237 Hymus Blvd.
Pointe Claire, Quebec
Canada, H9R 5C7
(514) 694-7710

Future Electronics
1000 St-Jean Babtiste #100
Quebec City, Quebec
Canada, G2E 5G5
(418) 877-6666

Marshall Industries
148 Brunswick Blvd.
Pointe Claire, Quebec
Canada, H9R 5B9
(514) 694-8142

GAL PRODUCT INDEX

Commercial Grade Devices

DEVICE	PINS	t_{PD} (ns)	I_{CC} (mA)	DESCRIPTION	PAGE
GAL16V8A/B	20	7.5, 10, 15, 25	55, 90, 115	E²CMOS Generic PLD	3-3
GAL20V8A/B	24	7.5, 10, 15, 25	55, 90, 115	E²CMOS Generic PLD	3-9
GAL18V10	20	15, 20	115	E²CMOS Universal PLD	3-15
GAL22V10/B	24	10, 15, 25	130	E²CMOS Universal PLD	3-19
GAL26CV12	28	15, 20	130	E²CMOS Universal PLD	3-25
GAL20RA10	24	12, 15, 20, 30	100	E²CMOS Asynchronous PLD	3-29
GAL20XV10B	24	10, 15	90	E²CMOS Exclusive-OR PLD	3-33
GAL6001	24	30, 35	150	E²CMOS FPLA	3-37
ispGAL16Z8	24	20, 25	90	E²CMOS In-System-Programmable PLD	3-41

Industrial Grade Devices

DEVICE	PINS	t_{PD} (ns)	I_{CC} (mA)	DESCRIPTION	PAGE
GAL16V8A/B	20	10, 15, 20, 25	65, 130	E²CMOS Generic PLD	See GAL Data Book
GAL20V8A/B	24	10, 15, 20, 25	65, 130	E²CMOS Generic PLD	See GAL Data Book
GAL18V10	20	20	125	E²CMOS Universal PLD	See GAL Data Book
GAL22V10/B	24	15, 20, 25	150	E²CMOS Universal PLD	See GAL Data Book
GAL26CV12	28	20	150	E²CMOS Universal PLD	See GAL Data Book
GAL20RA10	24	20	120	E²CMOS Asynchronous PLD	See GAL Data Book

MIL-STD-883C Grade Devices

DEVICE	PINS	t_{PD} (ns)	I_{CC} (mA)	DESCRIPTION	PAGE
GAL16V8A/B	20	10, 15, 20, 25, 30	65, 130	E²CMOS Universal PLD	See GAL Data Book
GAL20V8A	24	15, 20, 25, 30	65, 130	E²CMOS Universal PLD	See GAL Data Book
GAL22V10	24	15, 20, 25, 30	150	E²CMOS Universal PLD	See GAL Data Book
GAL26CV12	28	20, 25	160	E²CMOS Universal PLD	See GAL Data Book
GAL20RA10	24	20, 25	120	E²CMOS Asynchronous PLD	See GAL Data Book

Lattice Semiconductor Corp.
5555 Northeast Moore Ct.
Hillsboro, Oregon 97124 USA
Telephone: (503) 681-0118
FAX: (503) 681-3037

Printed on recycled paper

Order #: B0006