

**PHILIPS**

Semi-custom (PLD)

**IC13 1987**

**PHILIPS**

Data handbook



Electronic  
components  
and materials

Integrated circuits

Book IC13

1987

Semi-custom

Programmable Logic Devices (PLD)

## SEMI-CUSTOM PROGRAMMABLE LOGIC DEVICES

	<i>page</i>
Preface .....	.xi
Product status and definitions .....	xii
Contents .....	.xiii
<b>Section 1</b>	
Quality and reliability .....	1-3
Alphanumeric Index .....	1-9
Selection Guide .....	1-10
Ordering Information .....	1-11
Note to Users .....	1-12
<b>Section 2 – Introduction</b>	
What is Signetics Programmable Logic .....	2-3
Programming Information .....	2-48
<b>Section 3 – Advanced Product Preview</b>	
Index .....	3-1
<b>Section 4 – PLD Data Sheets</b>	
Series 20 .....	4-3
<b>Section 5 – PLD Data Sheets</b>	
Series 24 .....	5-3
<b>Section 6 – PLD Data Sheets</b>	
Series 28 .....	6-3
<b>Section 7 – Military Products</b>	
Military Errata .....	7-1
<b>Section 8 – Development Software</b>	
AMAZE – Automatic Map and Zap Equations Design Software .....	8-3
<b>Section 9 – Application Notes</b>	
Index .....	9-1
<b>Section 10 – Package Outlines</b>	
Index .....	10-1
Package Outlines .....	10-3



Signetics reserves the right to make changes, without notice, in the products, including circuits, standard cells, and/or software, described or contained herein in order to improve design and/or performance. Signetics assumes no responsibility or liability for the use of any of these products, conveys no license or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified. Applications that are described herein for any of these products are for illustrative purposes only. Signetics makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

**LIFE SUPPORT POLICY**

**SIGNETICS PRODUCTS ARE NOT FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT EXPRESS WRITTEN APPROVAL OF AN OFFICER OF SIGNETICS CORPORATION. As used herein:**

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

Signetics registers eligible circuits under  
the Semiconductor Chip Protection Act.

© Copyright 1986 Signetics Corporation

All rights reserved.

## DATA HANDBOOK SYSTEM

Our Data Handbook System comprises more than 60 books with specifications on electronic components, subassemblies and materials. It is made up of four series of handbooks:

ELECTRON TUBES	BLUE
SEMICONDUCTORS	RED
INTEGRATED CIRCUITS	PURPLE
COMPONENTS AND MATERIALS	GREEN

The contents of each series are listed on pages iv to vii.

The data handbooks contain all pertinent data available at the time of publication, and each is revised and reissued periodically.

When ratings or specifications differ from those published in the preceding edition they are indicated with arrows in the page margin. Where application information is given it is advisory and does not form part of the product specification.

Condensed data on the preferred products of Philips Electronic Components and Materials Division is given in our Preferred Type Range catalogue (issued annually).

Information on current Data Handbooks and on how to obtain a subscription for future issues is available from any of the Organizations listed on the back cover.

Product specialists are at your service and enquiries will be answered promptly.

## ELECTRON TUBES (BLUE SERIES)

The blue series of data handbooks comprises:

- T1 Tubes for r.f. heating**
- T2a Transmitting tubes for communications, glass types**
- T2b Transmitting tubes for communications, ceramic types**
- T3 Klystrons**
- T4 Magnetrons for microwave heating**
- T5 Cathode-ray tubes**  
Instrument tubes, monitor and display tubes, C.R. tubes for special applications
- T6 Geiger-Müller tubes**
- T8 Colour display systems**  
Colour TV picture tubes, colour data graphic display tube assemblies, deflection units
- T9 Photo and electron multipliers**
- T10 Plumbicon camera tubes and accessories**
- T11 Microwave semiconductors and components**
- T12 Vidicon and Newvicon camera tubes**
- T13 Image intensifiers and infrared detectors**
- T15 Dry reed switches**
- T16 Monochrome tubes and deflection units**  
Black and white TV picture tubes, monochrome data graphic display tubes, deflection units

## SEMICONDUCTORS (RED SERIES)

The red series of data handbooks comprises:

- S1 Diodes**  
Small-signal silicon diodes, voltage regulator diodes (< 1,5 W), voltage reference diodes, tuner diodes, rectifier diodes
- S2a Power diodes**
- S2b Thyristors and triacs**
- S3 Small-signal transistors**
- S4a Low-frequency power transistors and hybrid modules**
- S4b High-voltage and switching power transistors**
- S5 Field-effect transistors**
- S6 R.F. power transistors and modules**
- S7 Surface mounted semiconductors**
- S8a Light-emitting diodes**
- S8b Devices for optoelectronics**  
Optocouplers, photosensitive diodes and transistors, infrared light-emitting diodes and infrared sensitive devices, laser and fibre-optic components
- S9 Power MOS transistors**
- S10 Wideband transistors and wideband hybrid IC modules**
- S11 Microwave transistors**
- S12 Surface acoustic wave devices**
- S13 Semiconductor sensors**
- S14 Liquid Crystal Displays**

## INTEGRATED CIRCUITS (PURPLE SERIES)

The purple series of handbooks comprises:

<b>IC01</b>	<b>Radio, audio and associated systems</b> Bipolar, MOS	published 1986
<b>IC02a/b</b>	<b>Video and associated systems</b> Bipolar, MOS	published 1986
<b>IC03</b>	<b>Integrated circuits for telephony</b> Bipolar, MOS	published 1986
<b>IC04</b>	<b>HE4000B logic family</b> CMOS	published 1986
<b>IC05N</b>	<b>HE4000B logic family – uncased ICs</b> CMOS	published 1984
<b>IC06N</b>	<b>High-speed CMOS; PC74HC/HCT/HCU</b> Logic family	published 1986
<b>IC08</b>	<b>ECL 10K and 100K logic families</b>	published 1986
<b>IC09N</b>	<b>TTL logic series</b>	published 1986
<b>IC10</b>	<b>Memories</b> MOS, TTL, ECL	new issue 1987
<b>IC11N</b>	<b>Linear LSI</b>	published 1985
<b>Supplement to IC11N</b>	<b>Linear LSI</b>	published 1986
<b>IC12</b>	<b>I<sup>2</sup>C-bus compatible ICs</b>	not yet issued
<b>IC13</b>	<b>Semi-custom</b> Programmable Logic Devices (PLD)	new issue 1987
<b>IC14</b>	<b>Microcontrollers and peripherals</b> Bipolar, MOS	new issue 1987
<b>IC15</b>	<b>FAST TTL logic series</b>	published 1986
<b>IC16</b>	<b>CMOS integrated circuits for clocks and watches</b>	published 1986
<b>IC17</b>	<b>Integrated Services Digital Networks (ISDN)</b>	not yet issued
<b>IC18</b>	<b>Microprocessors and peripherals</b>	new issue 1987

## COMPONENTS AND MATERIALS (GREEN SERIES)

The green series of data handbooks comprises:

- C2** Television tuners, coaxial aerial input assemblies, surface acoustic wave filters
- C3** Loudspeakers
- C4** Ferroxcube potcores, square cores and cross cores
- C5** Ferroxcube for power, audio/video and accelerators
- C6** Synchronous motors and gearboxes
- C7** Variable capacitors
- C8** Variable mains transformers
- C9** Piezoelectric quartz devices
- C11** Varistors, thermistors and sensors
- C12** Potentiometers, encoders and switches
- C13** Fixed resistors
- C14** Electrolytic and solid capacitors
- C15** Ceramic capacitors
- C16** Permanent magnet materials
- C17** Stepping motors and associated electronics
- C18** Direct current motors
- C19** Piezoelectric ceramics
- C20** Wire-wound components for TVs and monitors
- C22** Film capacitors



## GENERAL CONTENTS

**Preface**  
**Product status definitions**  
**Contents**





### Application Specific Products

The continuing trend of system integration has created new challenges for Design Engineering. They must strive to consolidate higher complexity, and more feature intensive circuits into designs without sacrificing flexibility. Today's competitive electronic marketplace has created the need for logic devices, which can provide cost effective methods of reducing random logic requirements, interface with fixed and custom LSI logic, and maintain the flexible features required to make necessary system modifications prior to production. Signetics has responded to this need with Signetics Programmable Logic Devices (Signetics PLD).

The Signetics Programmable Logic family consists of a relatively few devices which are designed to address logic needs ranging from random gates in the case of the Field Programmable Gate Arrays, to highly sophisticated state machines in the case of Field Programmable Logic Sequencers. Signetics pioneered the fully-programmable AND/OR/INVERT architecture in 1975. In 1987, Signetics will introduce a new concept in PLD architectures. Moving away from an AND-OR array structure, the Signetics' Programmable Macro Logic architecture relies on a single NAND array concept. Signetics' novel architectural approach to Programmable Logic incorporates all the advantages associated with more dense complex gate arrays while retaining all the benefits of instantly programmable PLDs. By using the programmable features allowed by either powerful architecture, Signetics Programmable Logic can encompass wide-ranging levels of integration without the necessity of a multitude of devices, each with a different I/O configuration. The flexible programming structure allows the designer to "mold" the device architecture to the range of applications typically found in system design. In this way, one device can be used to perform several functions in the same design.

The 1987 Signetics Programmable Logic Data Manual contains information that the designer will require in order to effectively utilize these products.

Signetics Application Specific Products Marketing

### Application Specific Products

#### DEFINITIONS

<b>Data Sheet Identification</b>	<b>Product Status</b>	<b>Definition</b>
<i>Objective Specification</i>	<b>Formative or In Design</b>	This data sheet contains the design target or goal specifications for product development. Specifications may change in any manner without notice.
<i>Preliminary Specification</i>	<b>Preproduction Product</b>	This data sheet contains preliminary data and supplementary data will be published at a later date. Signetics reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.
<i>Product Specification</i>	<b>Full Production</b>	This data sheet contains Final Specifications. Signetics reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.

## Application Specific Products

Preface .....	iii
Product Status .....	iv
<b>Section 1</b>	
<b>Quality and Reliability</b> .....	1-3
<b>Alphanumeric Index</b> .....	1-9
<b>Selection Guide</b> .....	1-10
<b>Ordering Information</b> .....	1-11
<b>Note to Users</b> .....	1-12
<b>Section 2 – Introduction</b>	
<b>What Is Signetics' Programmable Logic</b> .....	2-3
<b>Programming Information</b> .....	2-48
<b>Section 3 – Advanced Product Preview</b>	
PLMC153      Mask-Programmable Logic Array (18 × 42 × 10) .....	3-3
PLUS153     Field-Programmable Logic Array (18 × 42 × 10) .....	3-5
PLUS173     Field-Programmable Logic Array (22 × 42 × 10) .....	3-7
PLHS501     Programmable Macro Logic Random Logic Unit (32 × 72 × 24) .....	3-9
PLHS502     Programmable Macro Logic Sync/Async Register Macro Unit (32 × 64 × 24) .....	3-19
<b>Section 4 – PLD Data Sheets</b>	
<b>Series 20</b>	
PLS151     Field-Programmable Gate Array (18 × 15 × 12) .....	4-3
PLS153     Field-Programmable Logic Array (18 × 42 × 10) .....	4-9
PLS153A   Field-Programmable Logic Array (18 × 42 × 10) .....	4-15
PLHS153   Field-Programmable Logic Array (18 × 42 × 10) .....	4-21
PLS155     Field-Programmable Logic Sequencer (16 × 45 × 12) .....	4-27
PLS157     Field-Programmable Logic Sequencer (16 × 45 × 12) .....	4-37
PLS159     Field-Programmable Logic Sequencer (16 × 45 × 12) .....	4-47
PLS159A   Field-Programmable Logic Sequencer (16 × 45 × 12) .....	4-57
PLHS18P8A Programmable AND Array Logic (18 × 72 × 8) .....	4-67
<b>Section 5 – PLD Data Sheets</b>	
<b>Series 24</b>	
PLS161     Field-Programmable Logic Array (12 × 48 × 8) .....	5-3
PLS162     Field-Programmable Address Decoder (16 × 5) .....	5-9
PLS163     Field-Programmable Address Decoder (12 × 9) .....	5-15
PLS167     Field-Programmable Logic Sequencer (14 × 48 × 6) .....	5-21
PLS167A   Field-Programmable Logic Sequencer; Enhanced Speed (14 × 48 × 6) .....	5-33
PLS168     Field-Programmable Logic Sequencer (12 × 48 × 8) .....	5-45
PLS168A   Field-Programmable Logic Sequencer; Enhanced Speed (12 × 48 × 8) .....	5-57
PLS173     Field-Programmable Logic Array (22 × 42 × 10) .....	5-69
PLS179     Field-Programmable Logic Sequencer (20 × 45 × 12) .....	5-75
PLHS473   Field-Programmable Logic Array (20 × 24 × 11) .....	5-85
PLC473     Erasable-Programmable Logic Array (20 × 24 × 11) .....	5-91
<b>Section 6 – PLD Data Sheets</b>	
<b>Series 28</b>	
PLS100/PLS101 Field-Programmable Logic Array (16 × 48 × 8) .....	6-3
PLS103     Field-Programmable Gate Array (16 × 9 × 9) .....	6-10
PLS105     Field-Programmable Logic Sequencer (16 × 48 × 8) .....	6-16
PLS105A   Field-Programmable Logic Sequencer (16 × 48 × 8) .....	6-27
PLUS405A   Field-Programmable Logic Sequencer (16 × 64 × 8) .....	6-38
<b>Section 7 – Military Products</b>	
<b>Military Errata</b> .....	7-1

# Contents

## Section 8 – Development Software

<b>AMAZE – Automatic Map and Zap Equations Design Software</b> .....	8-3
--	-----

## Section 9 – Application Notes

AN7	Single Chip Multiprocessor Arbiter .....	9-3
AN8	Introduction to Signetics Programmable Logic .....	9-15
AN10	4-Bit Binary-to-7 Segment Decoder .....	9-27
AN11	PLD Programmable Retriggerable One-Shot .....	9-51
AN12	Low Cost Programmer for PLD 20-Pin Series .....	9-55
AN13	Oscillator With PLS159 .....	9-71
AN14	Latches and Flip-Flops With PLS153 .....	9-76
AN15	PLS159 Primer .....	9-89
AN16	PLS159 as Modulo Counters .....	9-100
AN17	Electronic Combination Lock .....	9-107
AN18	Schmitt Trigger Using PLS153 and PLS159 .....	9-110
AN19	Electronic Dice .....	9-118
AN20	Single Pin Debouncer Using PLS153/159 .....	9-122
AN21	9-Bit Parity Generator/Checker With PLS153/153A .....	9-125
AN22	PLS151 Toolkit .....	9-131
AN23	PLS168/168A Primer .....	9-139
AN24	PLS173 as a 10-Bit Comparator, 74LS460 .....	9-152
AN25	A Simple Entry/Exit Meeting Device (PLS157) .....	9-158
AN26	PLHS18P8A Primer .....	9-164
AN27	PLHS473 Primer .....	9-172
AN28	High-Speed 12-Bit Tracking A/D Converter Using PLS179 .....	9-180
AN29	PLHS501 Programmable Macro Logic Primer .....	9-190

## Section 10 – Package Outlines

### Package Outlines

A	PLASTIC: Leaded Chip Carrier .....	10-3
F	CERAMIC: Dual-In-Line .....	10-5
FA	CERAMIC: Dual-In-Line With Quartz Window .....	10-7
N	PLASTIC: Dual-In-Line .....	10-9

Application Specific Products

### INDEX

<b>Section 1</b>	
Quality and Reliability .....	1-3
Alphanumeric Index .....	1-9
Selection Guide .....	1-10
Ordering Information .....	1-11
Note to Users .....	1-12



## Application Specific Products

### SIGNETICS PROGRAMMABLE LOGIC QUALITY

Signetics has put together winning processes for manufacturing Programmable Logic. Our standard is zero defects, and current customer quality statistics demonstrate our commitment to this goal.

The PLD's produced in the Application Specific Products Division must meet rigid criteria as defined by our design rules and as evaluated with a thorough product characterization and quality process. The capabilities of our manufacturing process are measured and the results evaluated and reported through our corporate-wide QA05 data base system. The SURE (Systematic Uniform Reliability Evaluation) program monitors the performance of our product in a variety of accelerated environmental stress conditions. All of these programs and systems are intended to prevent product-related problems and to inform our customers and employees of our progress in achieving zero defects.

### RELIABILITY BEGINS WITH THE DESIGN

Quality and reliability must begin with design. No amount of extra testing or inspection will produce reliable ICs from a design that is inherently unreliable. Signetics follows very strict design and layout practices with its circuits. To eliminate the possibility of metal migration, current density in any path cannot exceed  $2 \times 10^5$  amps/cm<sup>2</sup>. Layout rules are followed to minimize the possibility of shorts, circuit anomalies, and SCR type latch-up effects. Numerous ground-to-substrate connections are required to ensure that the entire chip is at the same ground potential, thereby precluding internal noise problems.

### PRODUCT CHARACTERIZATION

Before a new design is released, the characterization phase is completed to insure that the distribution of parameters resulting from lot-to-lot variations is well within specified limits. Such extensive characterization data

also provides a basis for identifying unique application-related problems which are not part of normal data sheet guarantees. Characterization takes place from  $-55^\circ\text{C}$  to  $+125^\circ\text{C}$  and at  $+10\%$  supply voltage.

### QUALIFICATION

Formal qualification procedures are required for all new or changed products, processes and facilities. These procedures ensure the high level of product reliability our customers expect. New facilities are qualified by corporate groups as well as by the quality organizations of specific units that will operate in the facility. After qualification, products manufactured by the new facility are subjected to highly accelerated environmental stresses to ensure that they can meet rigorous failure rate requirements. New or changed processes are similarly qualified.

### QA05 - QUALITY DATA BASE REPORTING SYSTEM

The QA05 data reporting system collects the results of product assurance testing on all finished lots and feeds this data back to concerned organizations where appropriate action can be taken. The QA05 reports EPQ (Estimated Process Quality) and AOQ (Average Outgoing Quality) results for electrical, visual/mechanical, hermeticity, and documentation audits. Data from this system is available on request.

### THE SURE PROGRAM

The SURE (Systematic Uniform Reliability Evaluation) program audits/monitors products from all Signetics' divisions under a variety of accelerated environmental stress conditions. This program, first introduced in 1964, has evolved to suit changing product complexities and performance requirements.

The SURE program has two major functions: Long-term accelerated stress performance audit and a short-term accelerated stress monitor. In the case of Bipolar Memory and Programmable Logic products, samples are

selected that represent all generic product groups in all wafer fabrication and assembly locations.

### THE LONG-TERM AUDIT

One-hundred devices from each generic family are subjected to each of the following stresses every eight weeks:

- High Temperature Operating Life:  $T_J = 150^\circ\text{C}$ , 1000 hours, static biased or dynamic operation, as appropriate (worst case bias configuration is chosen)
- High Temperature Storage:  $T_J = 150^\circ\text{C}$ , 1000 hours
- Temperature Humidity Biased Life:  $85^\circ\text{C}$ , 85% relative humidity, 1000 hours, static biased
- Temperature Cycling (Air-to-Air):  $-65^\circ\text{C}$  to  $+150^\circ\text{C}$ , 1000 cycles

### THE SHORT-TERM MONITOR

Every other week a 50-piece sample from each generic family is run to 168 hours of pressure pot (15psig,  $121^\circ\text{C}$ , 100% saturated steam) and 300 cycles of thermal shock ( $-65^\circ\text{C}$  to  $+150^\circ\text{C}$ ).

In addition, each Signetics assembly plant performs SURE product monitor stresses weekly on each generic family and molded package by pin count and frame type. Fifty-piece samples are run on each stress, pressure pot to 96 hours, thermal shock to 300 cycles.

### SURE REPORTS

The data from these test matrices provides a basic understanding of product capability, an indication of major failure mechanisms and an estimated failure rate resulting from each stress. This data is compiled periodically and is available to customers upon request.

Many customers use this information in lieu of running their own qualification tests, thereby eliminating time-consuming and costly additional testing.



## Quality and Reliability

### RELIABILITY ENGINEERING

In addition to the product performance monitors encompassed in the Programmable Logic SURE program, Signetics' Corporate and Division Reliability Engineering departments sustain a broad range of evaluation and qualification activities.

Included in the engineering process are:

- Evaluation and qualification of new or changed materials, assembly/wafer-fab processes and equipment, product designs, facilities and subcontractors.
- Device or generic group failure rate studies.
- Advanced environmental stress development.
- Failure mechanism characterization and corrective action/prevention reporting.

The environmental stresses utilized in the engineering programs are similar to those utilized for the SURE monitor; however, more highly-accelerated conditions and extended durations typify the engineering projects. Additional stress systems such as biased pressure pot, power-temperature cycling, and cycle-biased temperature-humidity, are also included in the evaluation programs.

### FAILURE ANALYSIS

The SURE Program and the Reliability Engineering Program both include failure analysis activities and are complemented by corporate, divisional and plant failure analysis departments. These engineering units provide a service to our customers who desire detailed failure analysis support, who in turn provide Signetics with the technical understanding of the failure modes and mechanisms actually experienced in service. This information is essential in our ongoing effort to accelerate and improve our understanding of product failure mechanisms and their prevention.

### ZERO DEFECTS PROGRAM

In recent years, United States industry has increasingly demanded improved product quality. We at Signetics believe that the customer has every right to expect quality products from a supplier. The benefits which are derived from quality products can be summed up in the words, *lower cost of ownership*.

Those of you who invest in costly test equipment and engineering to assure that incoming products meet your specifications have a special understanding of the cost of ownership. And your cost does not end there; you are also burdened with inflated inventories, lengthened lead times and more rework.

### SIGNETICS UNDERSTANDS CUSTOMERS' NEEDS

Signetics has long had an organization of quality professionals, inside all operating units, coordinated by a corporate quality department. This broad decentralized organization provides leadership, feedback, and direction for achieving a high level of quality. Special programs are targeted on specific quality issues. For example, in 1978 a program to reduce electrically defective units for a major automotive manufacturer improved outgoing quality levels by an order of magnitude.

In 1980 we recognized that in order to achieve outgoing levels on the order of 100PPM (parts per million), down from an industry practice of 10,000PPM, we needed to supplement our traditional quality programs with one that encompassed all activities and all levels of the company. Such unprecedented

low defect levels could only be achieved by contributions from all employees, from the R and D laboratory to the shipping dock. In short, from a program that would effect a total cultural change within Signetics in our attitude toward quality.

### QUALITY PAYS OFF FOR OUR CUSTOMERS

Signetics' dedicated programs in product quality improvement, supplemented by close working relationships with many of our customers, have improved outgoing product quality more than twenty-fold since 1980. Today, many major customers no longer test Signetics circuits. Incoming product moves directly from the receiving dock to the production line, greatly accelerating throughput and reducing inventories. Other customers have pared significantly the amount of sampling done on our products. Others are beginning to adopt these cost-saving practices.

We closely monitor the electrical, visual, and mechanical quality of all our products and review each return to find and correct the cause. Since 1981, over 90% of our customers report a significant improvement in overall quality (see Figure 1).

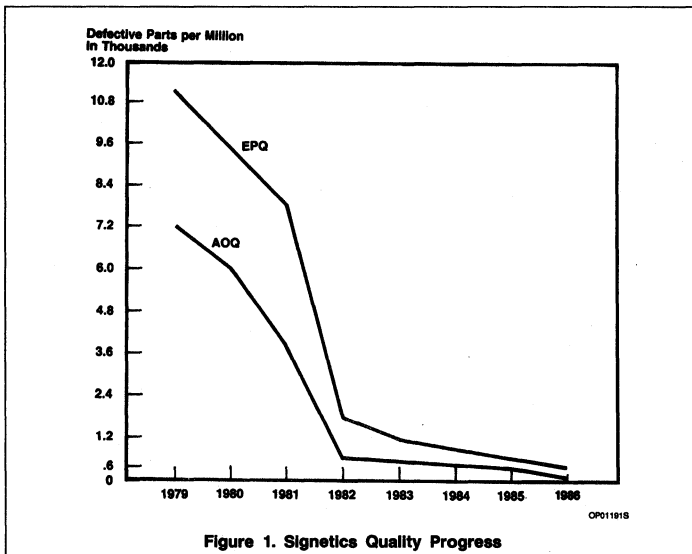


Figure 1. Signetics Quality Progress

## Quality and Reliability

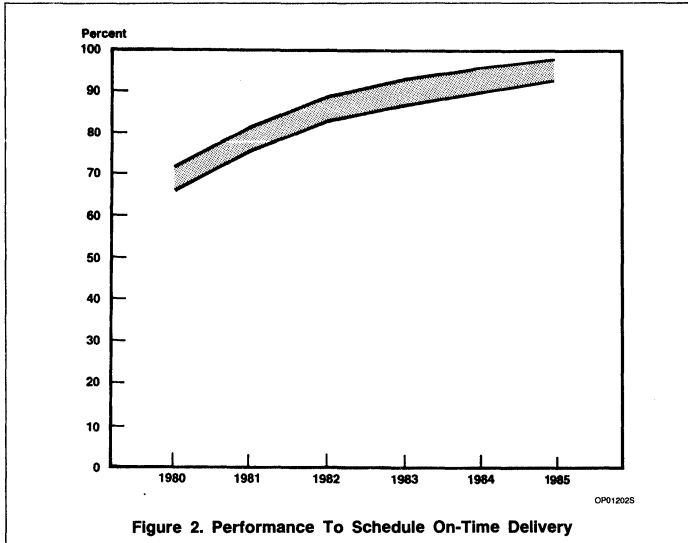


Figure 2. Performance To Schedule On-Time Delivery

At Signetics, quality means more than working circuits. It means on-time delivery of the right product at the agreed upon price (see Figure 2). Signetics considers Performance to Customer Request and Performance to Original Schedule Date to be key Quality issues. Employees treat delinquencies as quality defects. They analyze the cause for the delinquency and seek corrective action to prevent future occurrence. Continuous effort is given to try to achieve the ultimate goal of zero delinquencies.

### ONGOING QUALITY PROGRAM

The quality improvement program at Signetics is based on "Do it Right the First Time". The intent of this innovative program is to change the perception of Signetics' employees that somehow quality is solely a manufacturing issue where some level of defects is inevitable. This attitude has been replaced by one of acceptance of the fact that all errors and defects are preventable, a point of view shared by all technical and administrative functions equally.

This program extends into every area of the company, and more than 40 quality improvement teams throughout the organization drive its ongoing refinement and progress.

Key components of the program are the Quality College, the "Make Certain" Program, Corrective Action Teams, and the Error Cause Removal System.

The core concepts of doing it right the first time are embodied in the four absolutes of quality:

1. The definition of quality is conformance to requirements.
2. The system to achieve quality improvement is prevention.
3. The performance standard is zero defects.
4. The measurement system is the cost of quality.

### QUALITY COLLEGE

Almost continuously in session, Quality College is a prerequisite for all employees. The intensive curriculum is built around the four absolutes of quality; colleges are conducted at company facilities throughout the world.

### "MAKING CERTAIN" - ADMINISTRATIVE QUALITY IMPROVEMENT

Signetics' experience has shown that the largest source of errors affecting product and service quality is found in paperwork and in other administrative functions. The "Make Certain" program focuses the attention of management and administrative personnel on error prevention, beginning with each employee's own actions.

This program promotes defect prevention in three ways: by educating employees as to the impact and cost of administrative errors, by changing attitudes from accepting occasional errors to one of accepting a personal work standard of zero defects, and by providing a formal mechanism for the prevention of errors.

### CORRECTIVE ACTION TEAMS

Employees with the perspective, knowledge, and necessary skills to solve a problem are formed into ad hoc groups called Corrective Action Teams. These teams, a major force within the company for quality improvement, resolve administrative, technical and manufacturing problems.

### ECR SYSTEM (ERROR CAUSE REMOVAL)

The ECR System permits employees to report to management any impediments to doing the job right the first time. Once such an impediment is reported, management is obliged to respond promptly with a corrective program. Doing it right the first time in all company activities produces lower cost of ownership through defect prevention.

### PRODUCT QUALITY PROGRAM

To reduce defects in outgoing products, we created the Product Quality Program. This is managed by the Product Engineering Council, composed of the top product engineering and test professionals in the company. This group:

1. Sets aggressive product quality improvement goals;
2. provides corporate-level visibility and focus on problem areas;
3. serves as a corporate resource for any group requiring assistance in quality improvement; and
4. drives quality improvement projects.

As a result of this aggressive program, every major customer who reports back to us on product performance is reporting significant progress.

### VENDOR CERTIFICATION PROGRAM

Our vendors are taking ownership of their own product quality by establishing improved process control and inspection systems. They subscribe to the zero defects philosophy. Progress has been excellent.

## Quality and Reliability

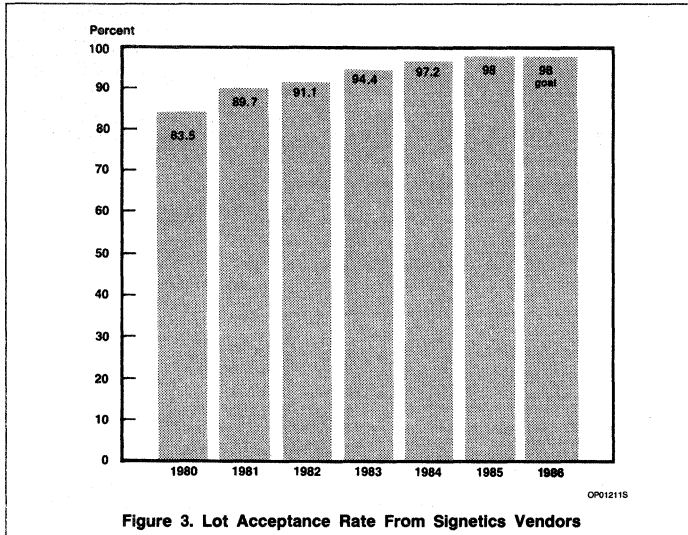


Figure 3. Lot Acceptance Rate From Signetics Vendors

Through intensive work with vendors, we have improved our lot acceptance rate on incoming materials as shown in Figure 3. Simultaneously, waivers of incoming material have been eliminated.

### MATERIAL WAIVERS

1986 - 0 (Goal)  
 1985 - 0  
 1984 - 0  
 1983 - 0  
 1982 - 2  
 1981 - 134

Higher incoming quality material ensures higher outgoing quality products.

### QUALITY AND RELIABILITY ORGANIZATION

Quality and reliability professionals at the divisional level are involved with all aspects of the product, from design through every step in the manufacturing process, and provide product assurance testing of outgoing product. A separate corporate-level group provides direction and common facilities.

#### Quality and Reliability Functions

- Manufacturing quality control
- Product assurance testing
- Laboratory facilities - failure analysis, chemical, metallurgy, thin film, oxides
- Environmental stress testing
- Quality and reliability engineering
- Customer liaison

### COMMUNICATING WITH EACH OTHER

For information on Signetics' quality programs or for any question concerning product quality, the field salesperson in your area will provide you with the quickest access to answers. Or, write on your letter-head directly to the corporate director of quality at the corporate address shown at the back of this manual.

We are dedicated to preventing defects. When product problems do occur, we want to know about them so we can eliminate their causes. Here are some ways we can help each other:

- Provide us with one informed contact within your organization. This will establish continuity and build confidence levels.
- Periodic face-to-face exchanges of data and quality improvement ideas between your engineers and ours can help prevent problems before they occur.
- Test correlation data is very useful. Line-pull information and field failure reports also help us improve product performance.
- Provide us with as much specific data on the problem as soon as possible to speed analysis and enable us to take corrective action.
- An advance sample of the devices in question can start us on the problem resolution before physical return of shipment.

This team work with you will allow us to achieve our mutual goal of improved product quality.

### MANUFACTURING: DOING IT RIGHT THE FIRST TIME

In dealing with the standard manufacturing flows, it was recognized that significant improvement would be achieved by "doing every job right the first time", a key concept of the quality improvement program. During the development of the program many profound changes were made. Figure 4, *Programmable Logic Process Flow*, shows the result. Key changes included such things as implementing 100% temperature testing on all products as well as upgrading test handlers to insure 100% positive binning. Some of the other changes and additions were to tighten the outgoing QA lot acceptance criteria to the tightest in the industry, with zero defect lot acceptance sampling across all three temperatures.

The achievements resulting from the improved process flow have helped Signetics to be recognized as the leading Quality supplier of Programmable Logic. These achievements have also led to our participation in several Ship-to-Stock programs, which our customers use to eliminate incoming inspection. Such programs reduce the user *cost of ownership* by saving both time and money.

### OUR GOAL: 100% PROGRAMMING YIELD

Our original goal back in the early 1970s was to develop a broad line of programmable products which would be recognized as having the best programming yield in the industry. Within the framework of a formal quality program, our efforts to improve circuit designs and refine manufacturing controls have resulted in major advances toward that goal.

Also within the framework of our formal quality program we have now established a stated goal of 100% programming yield. Through the increasing effectiveness of a quality attitude of "Do It Right The First Time" we're moving ever closer to that target.

Signetics PLD programming yields have been shown in collected data from internal audits and customer reporting to be consistently higher than comparable devices produced by our competition. We use systematic methods involving publication of exacting specifications of our programming algorithms, and through evaluation of those algorithms as implemented in industry standard programming equipment. Because of this we can assure our customers who program Signetics PLD's on such qualified equipment they will see consistently high yields. Our data base shows that average lot programming yield exceeds 97%.

# Quality and Reliability

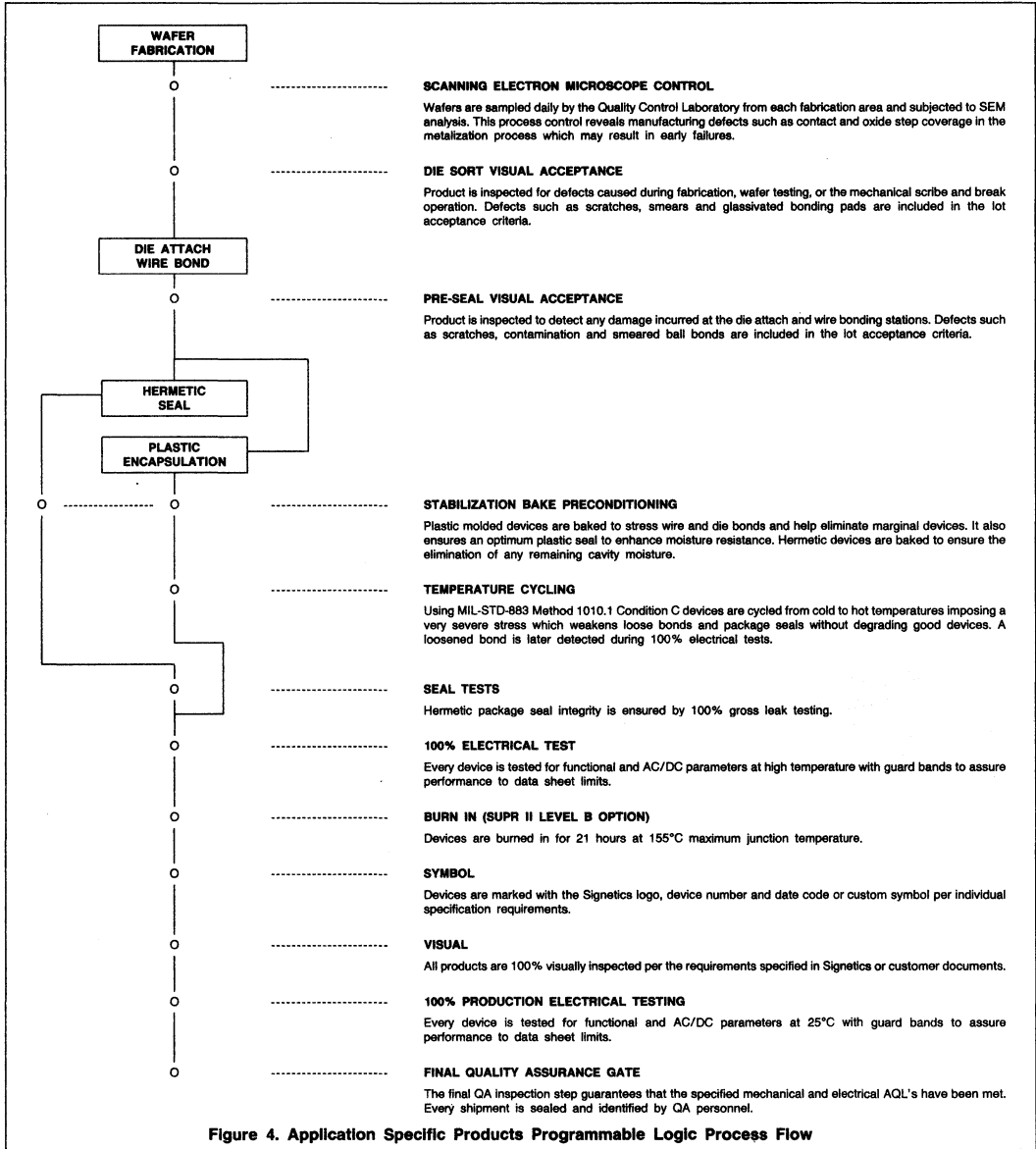


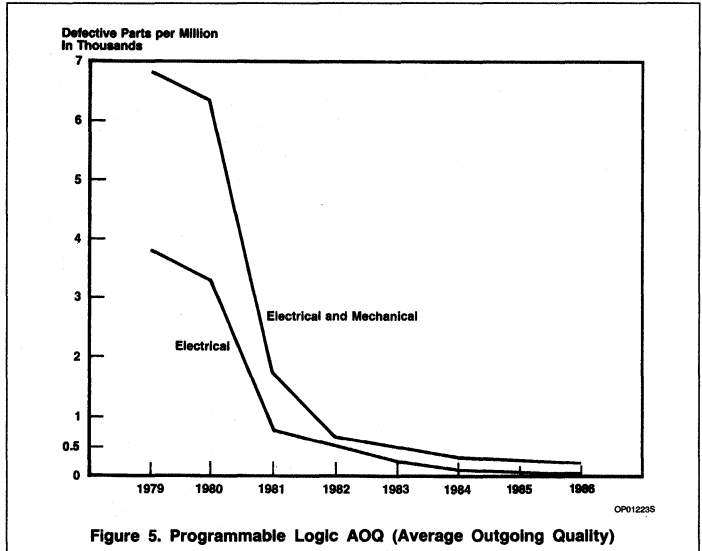
Figure 4. Application Specific Products Programmable Logic Process Flow

## Quality and Reliability

As time goes on the drive for a product line that has Zero Defects will grow in intensity. These efforts will provide both Signetics and our customers with the ability to achieve the mutual goal of improved product quality.

The Application Specific Quality Assurance department has monitored PPM progress, which can be seen in Figure 5. We are pleased with the progress that has been made, and expect to achieve even more impressive results as the procedures for accomplishing these tasks are fine tuned.

The real measure of any quality improvement program is the result that our customers see. The meaning of *Quality* is more than just working circuits. It means commitment to *On Time Delivery* at the *Right Place* of the *Right Quantity* of the *Right Product* at the *Agreed Upon Price*.



## Application Specific Products

<b>AN7</b>		Single Chip Multiprocessor Arbiter .....	9-3
<b>AN8</b>		Introduction To Signetics Programmable Logic .....	9-15
<b>AN10</b>		4-Bit Binary-To-7 Segment Decoder .....	9-27
<b>AN11</b>		PLD Programmable Retriggerable One-Shot .....	9-51
<b>AN12</b>		Low Cost Programmer For PLD 20-Pin Series .....	9-55
<b>AN13</b>		Oscillator With PLS159 .....	9-71
<b>AN14</b>		Latches & Flip-Flops With PLS153 .....	9-76
<b>AN15</b>		PLS159 Primer .....	9-89
<b>AN16</b>		PLS159 As Modulo Counters .....	9-100
<b>AN17</b>		Electronic Combination Lock .....	9-107
<b>AN18</b>		Schmitt Trigger Using PLS153 and PLS159 .....	9-110
<b>AN19</b>		Electronic Dice .....	9-118
<b>AN20</b>		Single Pin Debouncer Using 82S153/159 .....	9-122
<b>AN21</b>		9-Bit Parity Generator/Checker With 82S153/153A .....	9-125
<b>AN22</b>		PLS151 Toolkit .....	9-131
<b>AN23</b>		PLS168/168A Primer .....	9-139
<b>AN24</b>		PLS173 As A 10-Bit Comparator, 74LS460 .....	9-152
<b>AN25</b>		A Simple Entry/Exit Meeting Device (PLS157) .....	9-158
<b>AN26</b>		PLHS18P8A Primer .....	9-164
<b>AN27</b>		PLHS473 Primer .....	9-172
<b>AN28</b>		High-Speed 12-Bit Tracking A/D Converter Using PLS179 .....	9-180
<b>AN29</b>		PLHS501 Programmable Macro Logic Primer .....	9-191
<b>PLC473</b>	Series 24	Erasable-Programmable Logic Array (20 × 24 × 11) .....	5-91
<b>PLHS153</b>	Series 20	Field-Programmable Logic Array (18 × 42 × 10) .....	4-21
<b>PLHS473</b>	Series 24	Field-Programmable Logic Array (20 × 24 × 11) .....	5-85
<b>PLHS501</b>	Series 52	Programmable Macro Logic Random Logic Unit (32 × 72 × 24) .....	3-9
<b>PLHS502</b>	Series 52	Programmable Macro Logic Sync/Async Register Macro Unit (32 × 64 × 24) .....	3-19
<b>PLHS18P8A</b>	Series 20	Programmable AND Array Logic (18 × 72 × 8) .....	4-67
<b>PLMC153</b>	Series 20	Mask-Programmable Logic Array (18 × 42 × 10) .....	3-3
<b>PLS100/PLS101</b>	Series 28	Field-Programmable Logic Array (16 × 48 × 8) .....	6-3
<b>PLS103</b>	Series 28	Field-Programmable Gate Array (16 × 9 × 9) .....	6-10
<b>PLS105</b>	Series 28	Field-Programmable Logic Sequencer (16 × 48 × 8) .....	6-16
<b>PLS105A</b>	Series 28	Field-Programmable Logic Sequencer (16 × 48 × 8) .....	6-27
<b>PLS151</b>	Series 20	Field-Programmable Gate Array (18 × 15 × 12) .....	4-3
<b>PLS153</b>	Series 20	Field-Programmable Logic Array (18 × 42 × 10) .....	4-9
<b>PLS153A</b>	Series 20	Field-Programmable Logic Array (18 × 42 × 10) .....	4-15
<b>PLS155</b>	Series 20	Field-Programmable Logic Sequencer (16 × 45 × 12) .....	4-27
<b>PLS157</b>	Series 20	Field-Programmable Logic Sequencer (16 × 45 × 12) .....	4-37
<b>PLS159</b>	Series 20	Field-Programmable Logic Sequencer (16 × 45 × 12) .....	4-47
<b>PLS159A</b>	Series 20	Field-Programmable Logic Sequencer (16 × 45 × 12) .....	4-57
<b>PLS161</b>	Series 24	Field-Programmable Logic Array (12 × 48 × 8) .....	5-3
<b>PLS162</b>	Series 24	Field-Programmable Address Decoder (16 × 5) .....	5-9
<b>PLS163</b>	Series 24	Field-Programmable Address Decoder (12 × 9) .....	5-15
<b>PLS167</b>	Series 24	Field-Programmable Logic Sequencer (14 × 48 × 6) .....	5-21
<b>PLS167A</b>	Series 24	Field-Programmable Logic Sequencer; Enhanced Speed (14 × 48 × 6) .....	5-33
<b>PLS168</b>	Series 24	Field-Programmable Logic Sequencer (12 × 48 × 8) .....	5-45
<b>PLS168A</b>	Series 24	Field-Programmable Logic Sequencer; Enhanced Speed (12 × 48 × 8) .....	5-57
<b>PLS173</b>	Series 24	Field-Programmable Logic Array (22 × 42 × 10) .....	5-69
<b>PLS179</b>	Series 24	Field-Programmable Logic Sequencer (20 × 45 × 12) .....	5-75
<b>PLUS153</b>	Series 20	Field-Programmable Logic Array (18 × 42 × 10) .....	3-5
<b>PLUS173</b>	Series 24	Field-Programmable Logic Array (22 × 42 × 10) .....	3-7
<b>PLUS405A</b>	Series 28	Field-Programmable Logic Sequencer (16 × 64 × 8) .....	6-38

## Application Specific Products

SIGNETICS PN	ARCHITECTURE	PACKAGE	TOTAL INPUTS (# DEDICATED)	PRODUCT TERMS	INTERNAL STATE REGISTERS	OUTPUTS	T <sub>PD</sub>	I <sub>CC</sub>
						C, I/O, R, RI/O	(Typical Values) <sup>1,2</sup>	
	FPGA							
PLS103	16 × 9 × 9	28 Pin	16	9	0	9 C	20ns	120mA
PLS151	18 × 15 × 12	20 Pin	18 (6)	15	0	12 I/O	15ns	130mA
	FPAD							
PLS162	16 × 5	24 Pin	16	5	0	5 C	20ns	120mA
PLS163	12 × 9	24 Pin	12	9	0	9 C	20ns	120mA
	FPLA							
PLS100	16 × 48 × 8	28 Pin	16	48	0	8 C	35ns	120mA
PLS153	18 × 42 × 10	20 Pin	18 (8)	42	0	10 I/O	30ns	130mA
PLS153A	18 × 42 × 10	20 Pin	18 (8)	42	0	10 I/O	20ns	130mA
PLHS153**	18 × 42 × 10	20 Pin	18 (8)	42	0	10 I/O	15ns	150mA
PLUS153**	18 × 42 × 10	20 Pin	18 (8)	42	0	10 I/O	9ns	175mA
PLC153**	18 × 42 × 10	20 Pin	18 (8)	42	0	10 I/O	35ns	25mA
PLS161	12 × 48 × 8	24 Pin	12	48	0	8 C	35ns	120mA
PLS173	22 × 42 × 10	24 Pin	22 (12)	42	0	10 I/O	20ns	150mA
PLC173**	22 × 42 × 10	24 Pin	22 (12)	42	0	10 I/O	35ns	25mA
PLUS173**	22 × 42 × 10	24 Pin	22 (12)	42	0	10 I/O	9ns	175mA
PLHS473	20 × 24 × 11	24 Pin	20 (11)	24	0	2 C & 9 I/O	15ns	140mA
PLC473**	20 × 24 × 11	24 Pin	20 (11)	24	0	2 C & 9 I/O	30ns	25mA
PLHS18P8A	18 × 72 × 8	20 Pin	18 (10)	72	0	8 I/O	15ns	120mA
	FPLS							
PLS105	16 × 48 × 8	28 Pin	16	48	6	8 R	60ns	120mA
PLS105A	16 × 48 × 8	28 Pin	16	48	6	8 R	45ns	120mA
PLC105**	16 × 48 × 8	28 Pin	16	48	6	8 R	50ns	40mA
PLUS405**	16 × 64 × 8	28 Pin	16	64	8	8 R	15ns	160mA
PLS155	16 × 45 × 12	20 Pin	16 (4)	45	4*	8 I/O & 4 R I/O	55ns	150mA
PLS157	16 × 45 × 12	20 Pin	16 (4)	45	6*	6 I/O & 6 R I/O	55ns	150mA
PLS159	16 × 45 × 12	20 Pin	16 (4)	45	8*	4 I/O & 8 R I/O	55ns	150mA
PLS167	14 × 48 × 6	24 Pin	14	48	6 plus 2*	6 R	60ns	120mA
PLS167A	14 × 48 × 6	24 Pin	14	48	6 plus 2*	6 R	45ns	120mA
PLS168	12 × 48 × 8	24 Pin	12	48	6 plus 4*	8 R	60ns	120mA
PLS168A	12 × 48 × 8	24 Pin	12	48	6 plus 4*	8 R	45ns	120mA
PLS179	20 × 45 × 12	24 Pin	20 (8)	45	8*	4 I/O & 8 R I/O	45ns	145mA
	FPML							
PLHS501	32 × 72 × 24	52 Pin	32 (24)	72	0	16 C and 8 I/O	20ns	250mA
PLHS502	32 × 64 × 24	52 Pin	32 (24)	64	16	16 C or R and 8 C or R I/O	15/30ns	250mA

### OUTPUTS:

C = Combinatorial output

R = Registered output

I/O = Combinatorial I/O

R I/O = Registered I/O

\* State registers shared with output registers

\*\* Under development

### NOTES:

1. T<sub>PD</sub> = T<sub>IS</sub> + T<sub>CKO</sub> for registered devices

2. For worst case specifications refer to data sheets for individual product.

## Application Specific Products

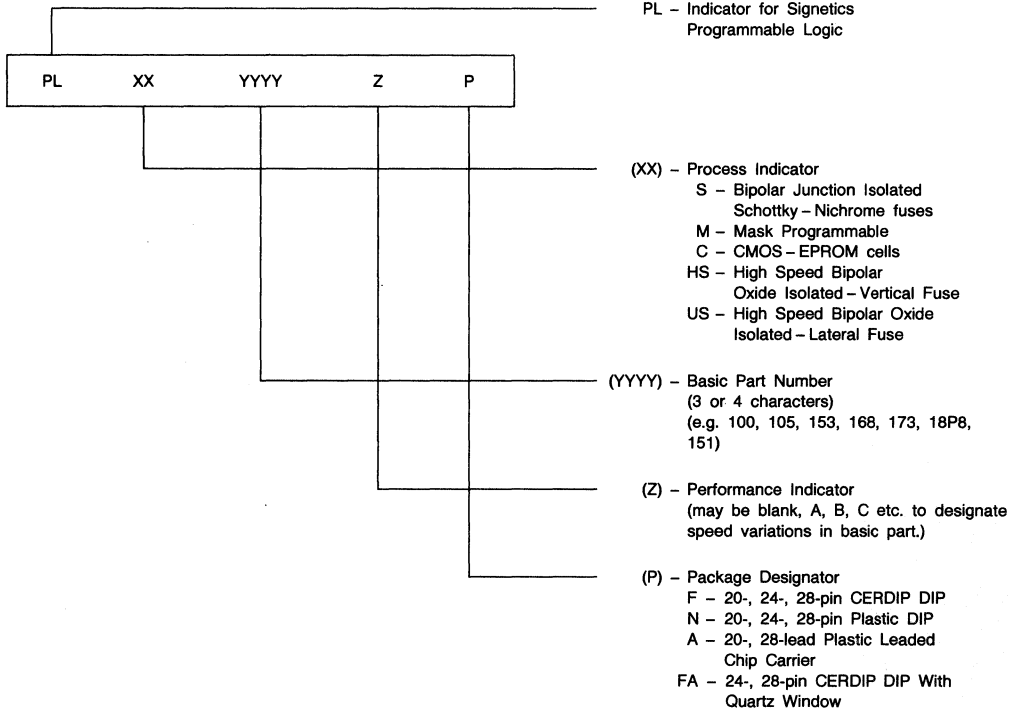
Signetics Programmable Logic Devices may be ordered by contacting either the local Signetics sales office, Signetics representatives or authorized distributors. A complete listing is located in the back of this manual.

Table 1 provides part number definition for Signetics PLD's. The Signetics part number system allows complete ordering information to be specified in the part number. The part number and product

description is located on each data sheet.

Military versions of these commercial products may be ordered. Please refer to the military products data manual for complete ordering information.

**Table 1. New Signetics PLD Part Numbering System**





### Application Specific Products

Signetics has changed its part numbering system for Programmable Logic Products. The change is simple and straightforward. Simply replace the old temperature code and "82" Bipolar Memory designator with "PL." This change in part numbering has been done to create a unique, identifiable image for Signetics Programmable Logic

Products. As we continue to introduce PLD's with new features, processes, and fusing technology, the new system will allow us to better describe these products through the product part number. In addition, more critical information can be placed on the device itself even when plastic leaded chip carriers are used. The new part number system does not

indicate any change in process or technology has occurred. The only thing changing is the nomenclature.

We sincerely hope this necessary change causes no inconveniences to our many customers. Thank you for your loyalty.

### PART NUMBERING EXAMPLES

OLD PART NUMBER	NEW PART NUMBER	DESCRIPTION
N82S100N	PLS100N	Signetics Programmable Logic Field Programmable Logic Array (FPLA) Bipolar Schottky Process - Nichrome Fuses - 28-Lead Dual-In-Line Package.
N82S105AA	PLS105AA	Signetics Programmable Logic Field Programmable Logic Sequencer (FPLS) Bipolar Schottky Process - Nichrome Fuses - High Speed Option - 28-Lead Plastic Leaded Chip Carrier.
N/A	PLC473FA	Signetics Programmable Logic Field Programmable Logic Array (FPLA) CMOS - UV Erasable Fuses - 24-Lead Cerdip Dual-In-Line Package with quartz window.
N/A	PLMC153F	Signetics Programmable Logic Programmable Logic Array CMOS - Mask Programmable - 20-Lead Cerdip Dual-In-Line Package.
N/A	PLHS18P8AA	Signetics Programmable Logic Field Programmable Array Oxide Isolated, Bipolar-Vertical Fuses - High Speed Option - 20-Pin Plastic Leaded Chip Carrier.

This change affects commercial product only.

### INDEX

#### Section 2 - Introduction

What is Signetics' Programmable Logic .....	2-3
Programming Information .....	2-48



#### Application Specific Products

#### WHAT IS PROGRAMMABLE LOGIC

In 1975, Signetics Corporation developed a new product family by combining its expertise in semi-custom gate array products and fuse-link Programmable Read Only Memories (PROMs). Out of this marriage came Signetics Programmable Logic Family. The PLS100 Field-Programmable Logic Array (FPLA) was the first member of this family. The FPLA was an important industry first in two ways. First, the AND/OR/INVERT architecture allowed the custom implementations of Sum of Product logic equations. Second, the three-level fusing allows complete flexibility in the use of this device family. All logic interconnections from input to output are programmable.

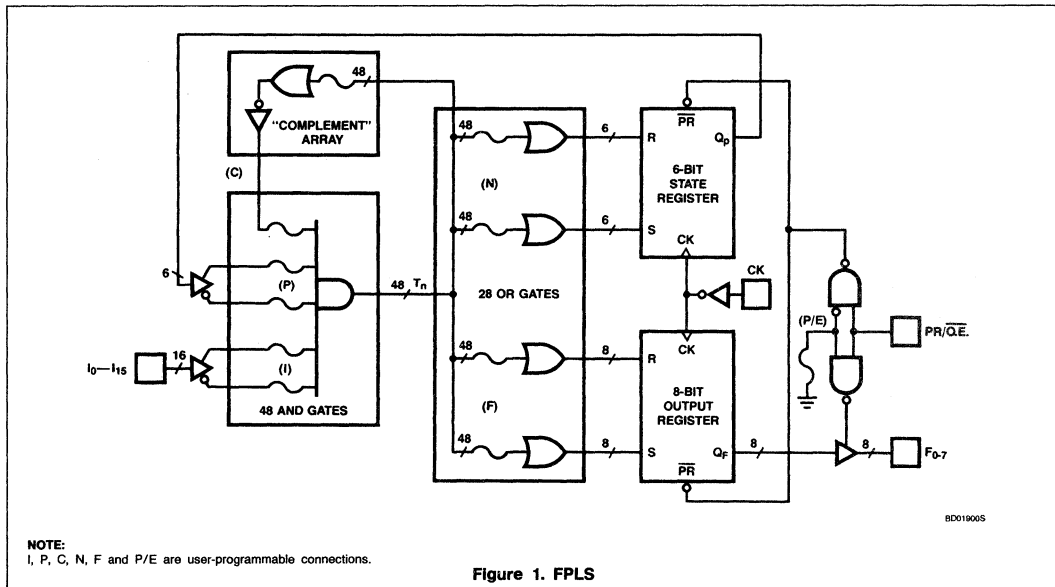
Development of this family did not stop with the PLS100. In 1977, the PLS103 Field-Programmable Gate Array (FPGA) and the PLS107 Field-Programmable ROM Patch (FPRP) were introduced.

The PLS105 Field-Programmable Logic Sequencer (FPLS) was announced in 1979. This device represents a significant step forward for PLD. The FPLS is a fully-implemented Mealy State Machine on a chip. Incorporated into its architecture are 48 P-terms, an 8-bit Output Register, and a 6-bit internal State Register. Reference Figure 1.

The FPLS can synchronously perform sequential routines at 20MHz. All of these products are now known as the Signetics PLD Series 28 Programmable Logic Family.

Signetics' next innovation in this area was the Series 20 Programmable Logic Family. All members of this family are assembled in 20-pin packages. While reducing the number of pins in this family, Signetics has utilized controlled I/O in order to maintain the utility of the Series 20. Appropriate control terms have been included to allow active control of pin direction. Reference Figure 2.

The latest addition to the Signetics Programmable Logic families is the Series 24 devices. The PLS161 FPLA and the PLS167 FPLS are 24-pin devices comparable to the 28-pin PLS100 and PLS105 respectively. These devices have been followed by the PLS162/PLS163 Field-Programmable Address Decoder and the PLS168 FPLS.



## Signetics Programmable Logic

## Introduction

Table 1. PLD Product Family

20-PIN, SERIES 20		
PART NUMBER	TYPE	CONFIGURATION
PLS151	FPGA	18-Input/12-Output
PLS153/153A/PLHS153	FPLA	18-Input/10-Output – 42-Term
PLS155-159	FPLS	16-Input/12-Output – 45-Term
PLS155	FPLS	4 Registered Outputs
PLS157	FPLS	6 Registered Outputs
PLS159	FPLS	8 Registered Outputs
PLHS18P8A	FPAF	18-Input/10-Output – 72-Term
24-PIN, SERIES 24		
PLS161	FPLA	12-Input/8-Output – 48-Term
PLS162	FPGA	16-Input/5-Output
PLS163	FPGA	12-Input/9-Output
PLS167	FPLS	14-Input/6-Output – 48-Term
		8-Bit State Registers
		6-Output Registers
PLS168	FPLS	12-Input/8-Output – 48-Term
		10-Bit State Registers
		8-Output Registers
PLS173/PLUS173	FPLA	22-Input/10-Output – 42-Term
PLS179	FPLS	16-Input/12-Output – 42-Term
		8-Bit State and Output Registers
PLHS473/PLC473	FPLA	20-Inputs/11-Output – 24-Term
28-PIN, SERIES 28		
PLS100/101	FPLA	16-Inputs/8-Output – 48-Term
PLS103	FPGA	16-Inputs/9-Output
PLS105/105A/PLC105	FPLS	16-Inputs/8-Output – 48-Term
		6-Bit State Register
		8-Output Registers
PLUS405	FPLS	16-Inputs/8-Outputs – 64-Terms
		8-Bit State and Output Registers

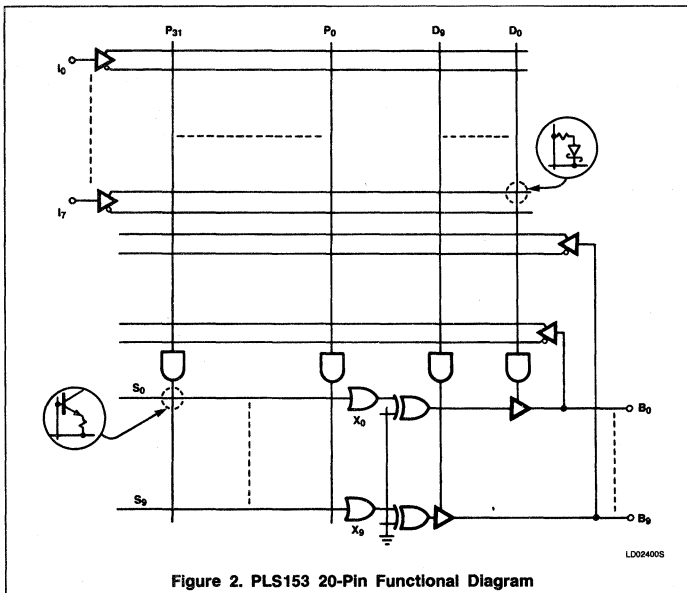
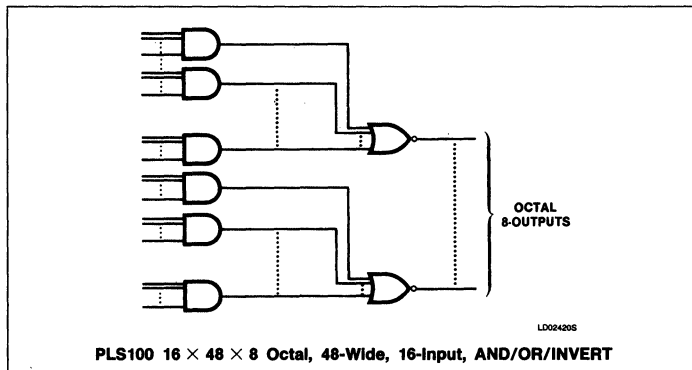
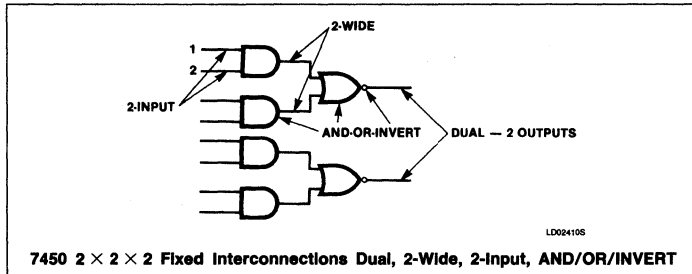


Figure 2. PLS153 20-Pin Functional Diagram



**PRODUCT DESCRIPTION**

While all PLD devices are architecturally similar, the Field-Programmable Logic Array is most representative of the concepts involved. The FPLA is an AND/OR/INVERT device. The FPLA is an AND/OR/INVERT device, with all internal interconnections programmable via fuse links. If we compare the PLS100 FPLA to the familiar 7450 AOI gate, the similarity of function becomes apparent.

**Package-Gate Replacement Potential**

All Signetics PLD devices are capable of replacing multiple discrete logic devices. The number of packages is dependent on the application and the device used.

If all product terms and inputs on the PLS100 are utilized, a total of 272 gates and 140 I.C.s can theoretically be replaced. (See Table 2.) Since the PLS100 is made up of logic structures which are not available as commercial logic devices, such as 32 input AND gates and 48 input OR gates, these numbers are based on breakdowns of these high complexity logic structures into structures which are available as discrete logic.

The typical application does not fully utilize the FPLA, however. We can assume that the typical application is as defined in Table 3.

The replacement of 15-20 I.C.s with one PLS100 is not unusual.

**Table 2. Gate Replacement (All Terms Used)**

	Gates	I.C.s
Each P-Term = 2 8-Input AND Gates & 1 2-Input AND Gate =	144	108
OR Matrix = 16 4-Input OR Gates =	128	32
	272	140

**Table 3. Typical Application**

4 Variables/P-Term	40 Gates
4 P-Terms/Output	15-20 I.C.s

# Signetics Programmable Logic

# Introduction

## PLD ECONOMICS

The reason any product line exists is because it solves a problem in a cost-effective manner.

PLD is no different in this respect. Let's assume that one PLS100 is replacing only 10 TTL logic packages.

If we assume that the average selling price of an SSI level device is \$0.25, then we have a basis for comparison.

- 10 I.C.s @ 0.35 = \$3.50
- 1 PLS100 FPLA = \$7.00

Considering piece-part cost alone does not fare well for the PLS100 FPLA. But, part cost is one of the smaller costs in systems manufacturing. Let's next consider PC board costs.

## PC BOARD

The purchase price of a PCB is calculated on a per square inch basis. Typical costs at the 5K level are \$0.23/in<sup>2</sup>, the cost for LS is 1.2in<sup>2</sup> = \$0.23/in<sup>2</sup> = \$0.28. For a 28-pin FPLA device, the packing density is

1-PLS100 per 2.5in<sup>2</sup>, giving a cost of 2.5in<sup>2</sup> × \$0.23/in<sup>2</sup> = \$0.58.

## PCB FABRICATION

Now let's consider the cost of building the board. The primary cost for PCB fabrication is parts preparation and stuffing.

Estimates by PCB manufacturers indicate a nominal cost of \$0.14 per device for LS TTL and \$0.16 for 28-pin FPLA. There is also a basic cost of \$0.02 per in<sup>2</sup> for handling, soldering, and inspecting.

The cost of incoming inspection and inventory/usage must also be considered in any fair comparison.

## INCOMING INSPECTION

The cost of incoming inspection can be calculated on a per-pin basis. Assuming an equal mix of 14- and 16-pin SSI parts and a testing cost of \$0.01 per pin, the LS TTL cost is \$0.15 per I.C. The corresponding cost for

the PLS100 would be \$0.01 × 28 or \$0.28 total.

## COMPONENT INVENTORY/USAGE

The inventory cost for LS TTL parts is dominated by inventory maintenance, while the LSI type parts, the piece price dominates. Assuming a 2% usage rate due to parts breakage, burnout, etc., in the manufacturing process, the parts cost for LS TTL is 2% × \$0.25 = \$0.005. Adding \$0.02 per device for inventory maintenance, the total LS cost \$0.03. For FPLA, 2% = \$7.00 ± \$0.02 = \$0.16.

Consideration must also be given to testing the completed PCB and Rework costs.

## PCB TESTING

Testing costs are generally figured on a PCB basis and are usually go-no-go tests using a "bed of nails". For this comparison, assume a PCB of 5" × 7". The cost for testing the completed PCB is estimated to be \$3.60 for 20 I.C.s. This breaks down as \$0.18 per LS TTL device or about \$0.45 per FPLA device.

## PCB REWORK

The main causes of rework are poorly plated holes, solder bridges, and parts inserted incorrectly. The "bed of nails" test can usually detect all of the faults. Rework costs about 50% of test costs. The result is \$1.80 per PCB, or \$0.09 per device for LS TTL and \$0.18 for the 28-pin FPLA.

System manufacturing costs are harder to define. For this comparison we estimated \$0.28 for each LS device and about \$0.65 for the FPLA. These costs include backplane wiring, final system assembly and test costs.

Now using these figures, the comparison in Table 4 can be made.

Table 4. Cost Analysis

	LS TTL	FPLA
<b>IC COMPONENTS</b>	.35	7.00
<b>PCB MANUFACTURING</b>		
Incoming Inspection	.15	.28
Component Inventory/Usage	.03	.16
PC Board	.28	.58
Fabrication	.14	.16
Test	.18	.45
Rework	.09	.18
<b>ASSEMBLY LABOR</b>	.28	.65
<b>COST PER COMPONENT</b>	1.50	9.46
<b>NUMBER OF COMPONENTS</b>	<u>× 10</u>	<u>× 1</u>
<b>TOTAL</b>	15.00	9.46
<b>CONCLUSION:</b>		
THE FPLA is a 37% savings over the LS TTL DESIGN.		

**PLD LOGIC SYNTHESIS**

**28-PIN**

PLD is capable of Logic Synthesis. No intermediate step is required to implement Bool-

ean Logic Equations with PLD. Each term in each equation simply becomes a direct entry into the Logic Program Table. The following example illustrates this straightforward concept:

$$X_0 = AB + \bar{C}D + B\bar{D}$$

$$\bar{X}_1 = \bar{A}B + \bar{C}D + EFG$$

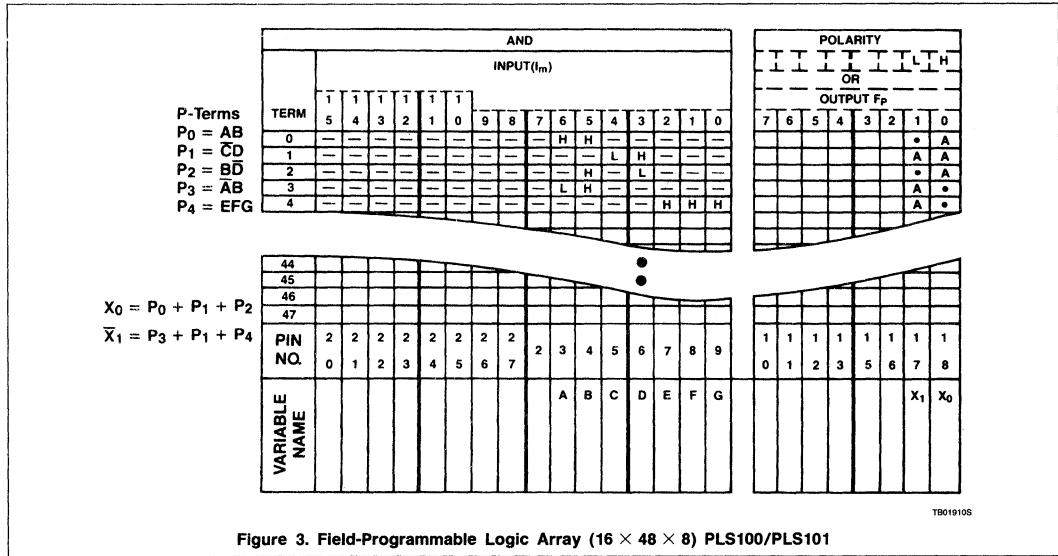


Figure 3. Field-Programmable Logic Array (16 × 48 × 8) PLS100/PLS101

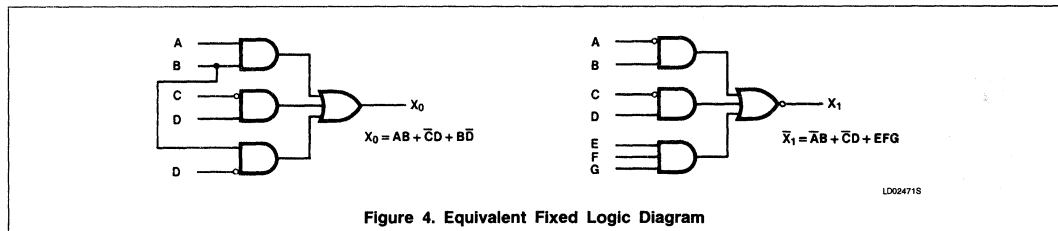


Figure 4. Equivalent Fixed Logic Diagram



# Signetics Programmable Logic

# Introduction

In the previous example, the two Boolean Logic equations were broken into Product terms. Each P-term was then programmed into the P-term section of the PLS100 FPLA

Program Table. This was accomplished in the following manner:

**Step 1**  
**Select which input pins I<sub>0</sub> - I<sub>15</sub> will correspond to the input variables.** In this case

A - G are the input variable names. I<sub>6</sub> through I<sub>10</sub> were selected to accept inputs A - G respectively.

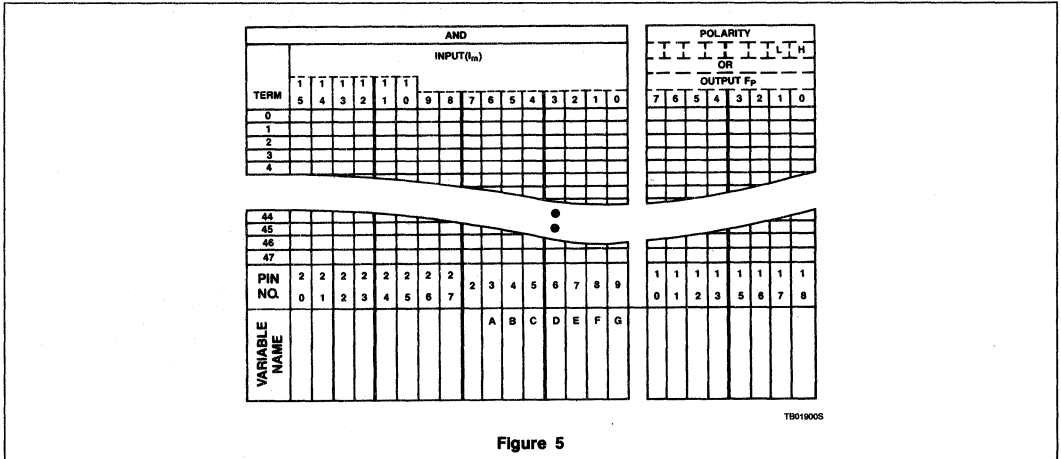


Figure 5

**Step 2**  
**Transfer the Boolean Terms to the FPLA Program Table.** This is done simply by defining each term and entering it on the Program Table.

e.g., P<sub>0</sub> = AB

This P-term translates to the Program Table by selecting A = I<sub>6</sub> = H and B = I<sub>5</sub> = H and entering the information in the appropriate column.

$$P_1 = \bar{C}D$$

This term is defined by selecting C = I<sub>4</sub> = L and D = I<sub>3</sub> = H, and entering the data into the Program Table. Continue this operation until all P-terms are entered into the Program Table.

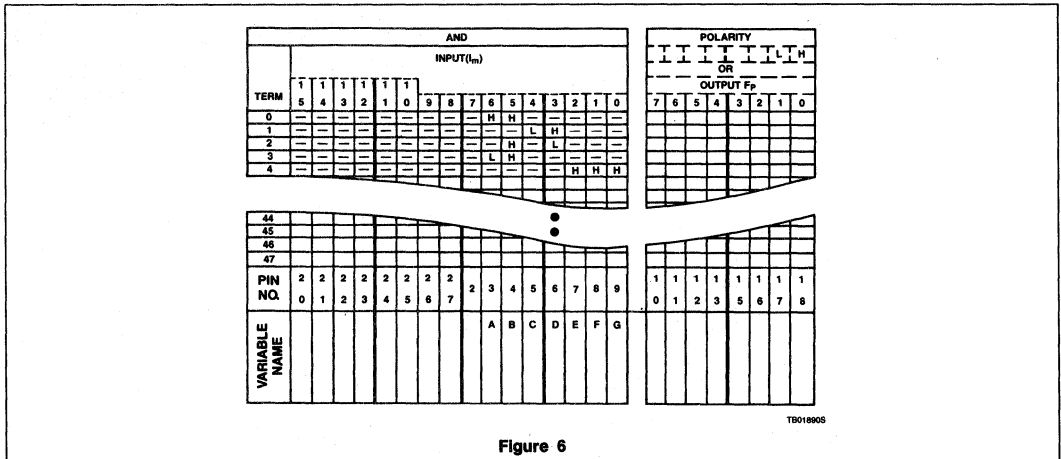


Figure 6

**Step 3**

Select which output pins correspond to each output function. In this case  $F_0 = \text{Pin } 18 = X_0$ , and  $F_1 = \text{Pin } 17 = X_1$ .

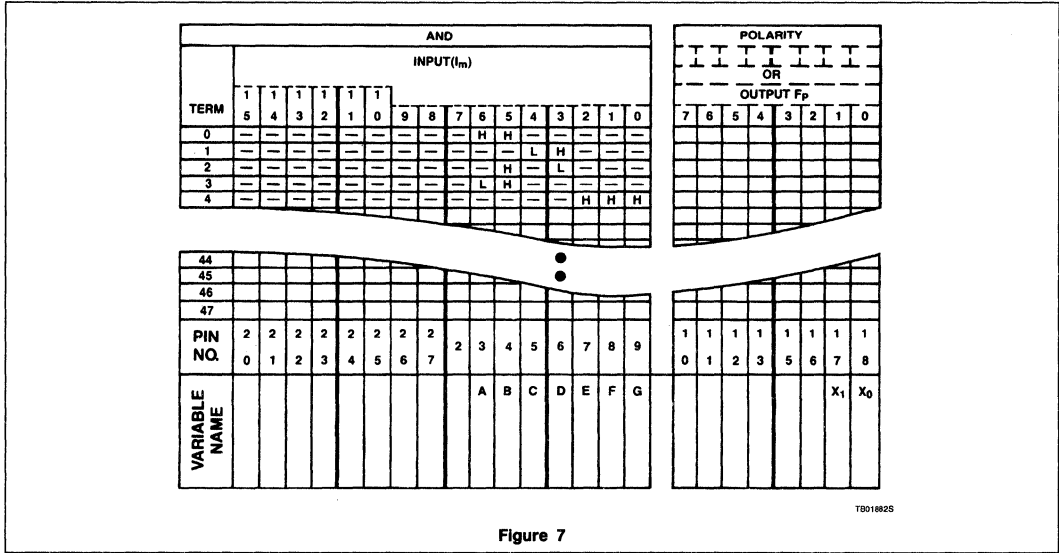


Figure 7

TB010025

**Step 4**

Select the Output Active Level desired for each Output Function. For  $X_0$  the active

level is high for a positive logic expression of this equation. Therefore, it is only necessary to place an (H) in the Active Level box above Output Function 0, ( $F_0$ ). Conversely,  $X_1$  can

be expressed as  $\bar{X}_1$  by placing an (L) in the Active Level box above Output Function 1, ( $F_1$ ).

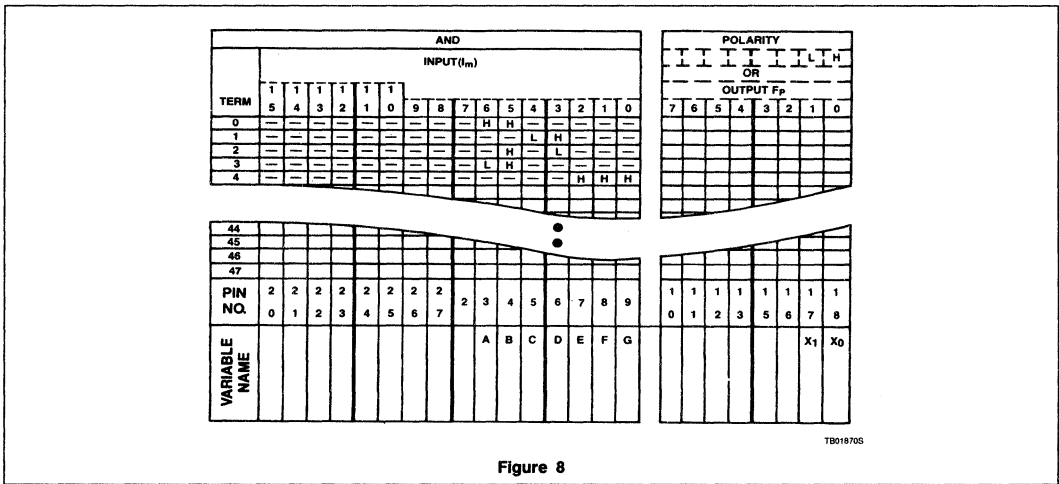


Figure 8

TB010705

# Signetics Programmable Logic

# Introduction

### Step 5

Select the P-Terms you wish to make active for each Output Function. In this case  $X_0 = P_0 + P_1 + P_2$ , so an A has been placed in the intersection box for  $P_0$  and  $X_0$ ,  $P_1$  and  $X_0$  and  $P_2$  and  $X_0$ .

Terms which are not active for a given output are made inactive by placing a (●) in the box under that P-term. Leave all unused P-terms unprogrammed.

Continue this operation until all outputs have been defined in the Program Table.

### Step 6

Enter the data into a Signetics approved programmer. The input format is identical to the Signetics Program Table. You specify the P-terms, Output Active Level, and which P-terms are active for each output exactly the way it appears on the Program Table.

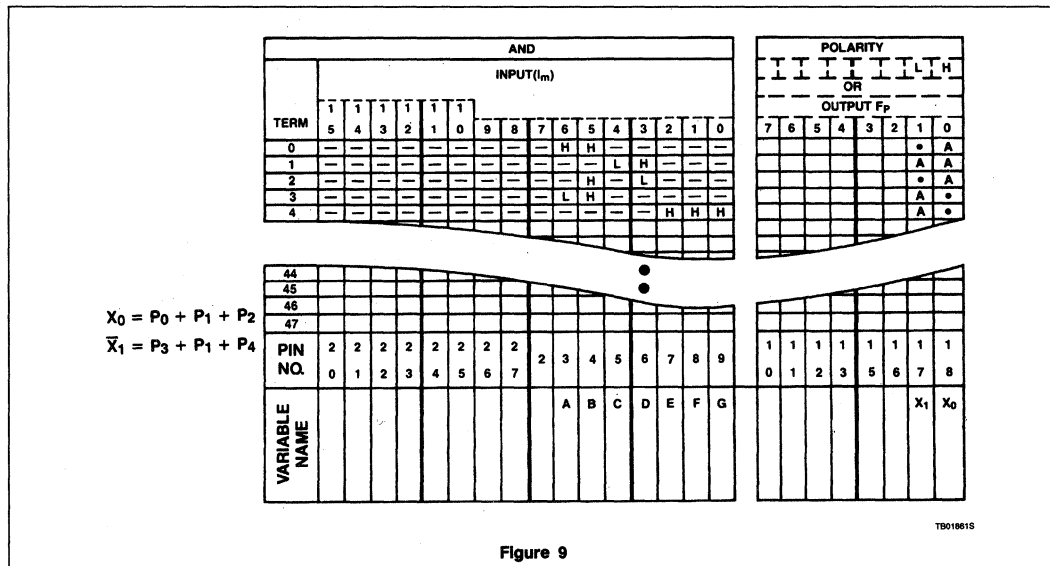


Figure 9

**PLD LOGIC SYNTHESIS**

**20-PIN**

When fewer inputs and outputs are required in a logic design and low cost is most important, the Signetics 20-pin PLD should be

considered first choice. The PLS153 is an FPLA with 8 inputs, 10 I/O pins, and 42 product terms. The user can configure the device by defining the direction of the I/O pins. This is easily accomplished by using the direction control terms  $D_0 - D_9$  to establish

the direction of pins  $B_0 - B_9$ . The D-terms control the Tri-state buffers found on the outputs of the EX-OR gates. Figures 10 and 11 show how the D-term configures each  $B_x$  pin.

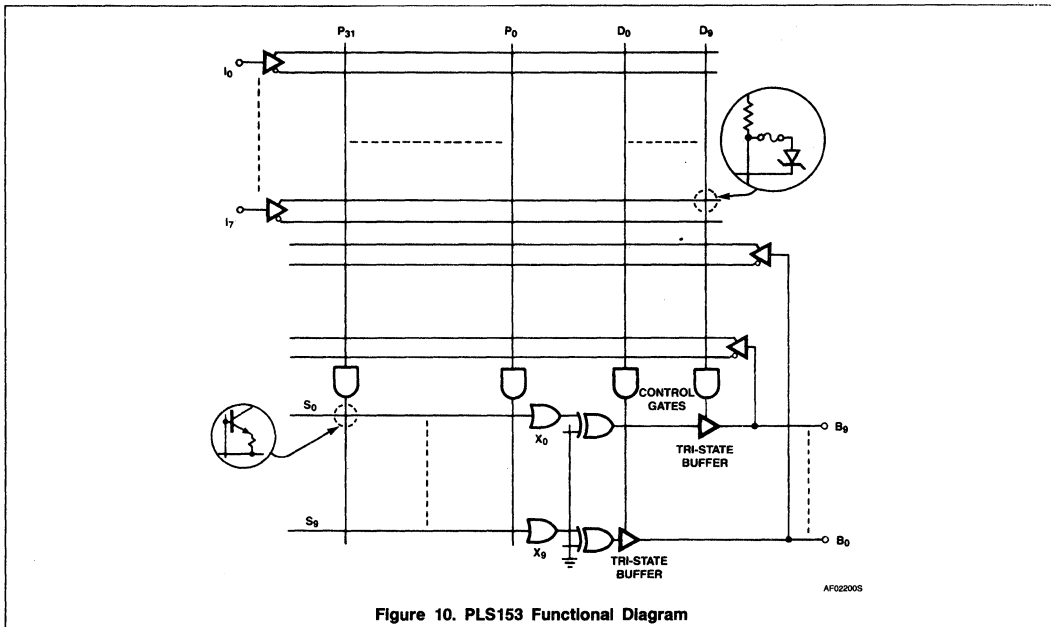


Figure 10. PLS153 Functional Diagram

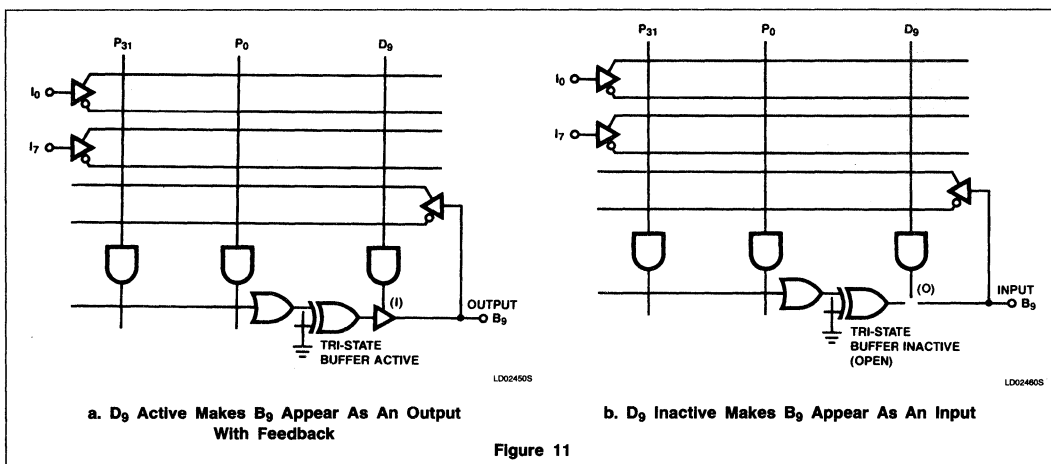


Figure 11

# Signetics Programmable Logic

# Introduction

To control each D-term, it is necessary to understand that each control gate is a 36-input AND gate. To make the Tri-state buffer active ( $B_x$  pin an output), the output of the control gate must be at logic HIGH (1). This can be accomplished in one of two ways. A

HIGH can be forced on all control gate input nodes, or fuses can be programmed. When a fuse is programmed, that control gate input node is internally pulled up to HIGH (1). See Figure 12 and Figure 13.

Programming the fuse permanently places a HIGH (1) on the input to the control gate. The input pin no longer has any effect on that state.

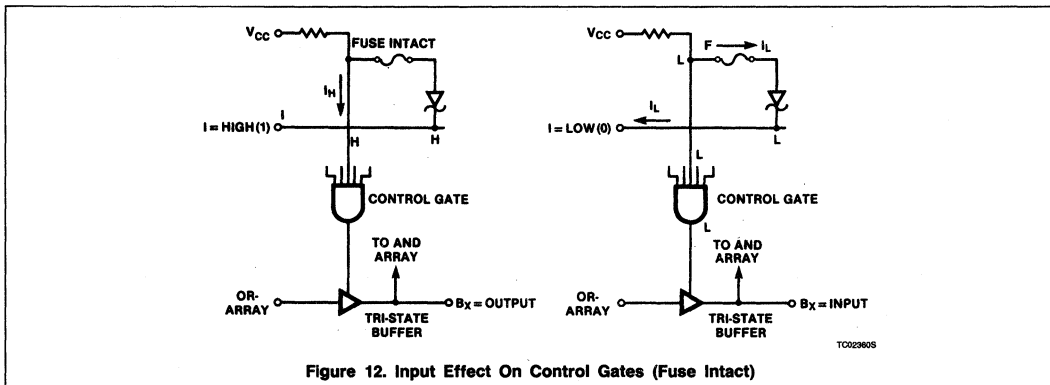


Figure 12. Input Effect On Control Gates (Fuse Intact)

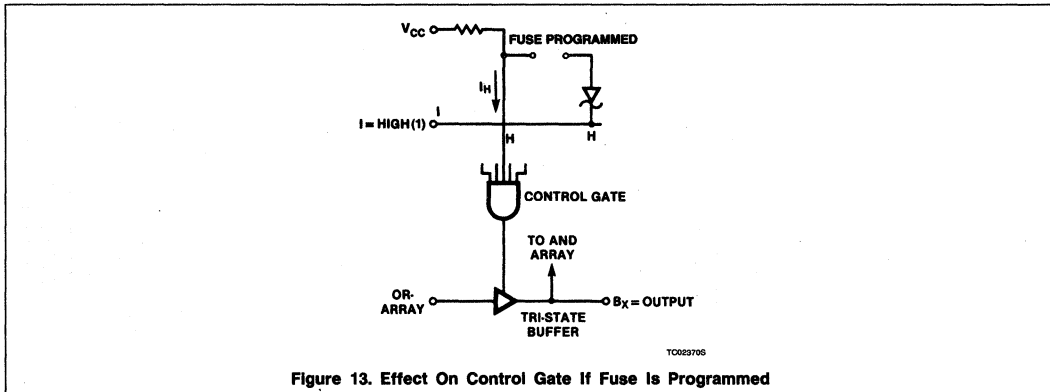


Figure 13. Effect On Control Gate If Fuse Is Programmed

# Signetics Programmable Logic

# Introduction

## DEDICATING B<sub>x</sub> PIN DIRECTION

Since each input to the D-terms is true and complement buffered (see Figure 11), when the device is shipped with all fuses intact, all control gates have half of the 36 input lines at logic low (0). The result of this is all Control Gate outputs are low (0) and the Tri-state buffers are inactive. This results in all B<sub>x</sub> pins being in the input condition. The resultant

device is, therefore, an 18-input, 0-output FPLA. While useful as a bit bucket or Write-Only-Memory (WOM), most applications require at least one output. Clearly, the first task is to determine which of the B<sub>x</sub> pins are to be outputs. The next step is to condition the control gate to make the Tri-state buffer for those gates active. To dedicate B<sub>0</sub> and B<sub>1</sub> as outputs, it is necessary to program all fuses to the inputs to Control Gates D<sub>0</sub> and D<sub>1</sub>. This internally pulls all inputs to those gates to

HIGH (1) permanently. Since all inputs to the Control Gates are HIGH (1), the output is HIGH (1) and the Tri-state buffers for B<sub>0</sub> and B<sub>1</sub> are active. This permanently enables B<sub>0</sub> and B<sub>1</sub> as outputs. Note that even though B<sub>0</sub> and B<sub>1</sub> are outputs, the output data is available to the AND array via the internal feedback (see Figure 11a).

To program this data, the PLS153 Program Table is used as shown in Figure 14.

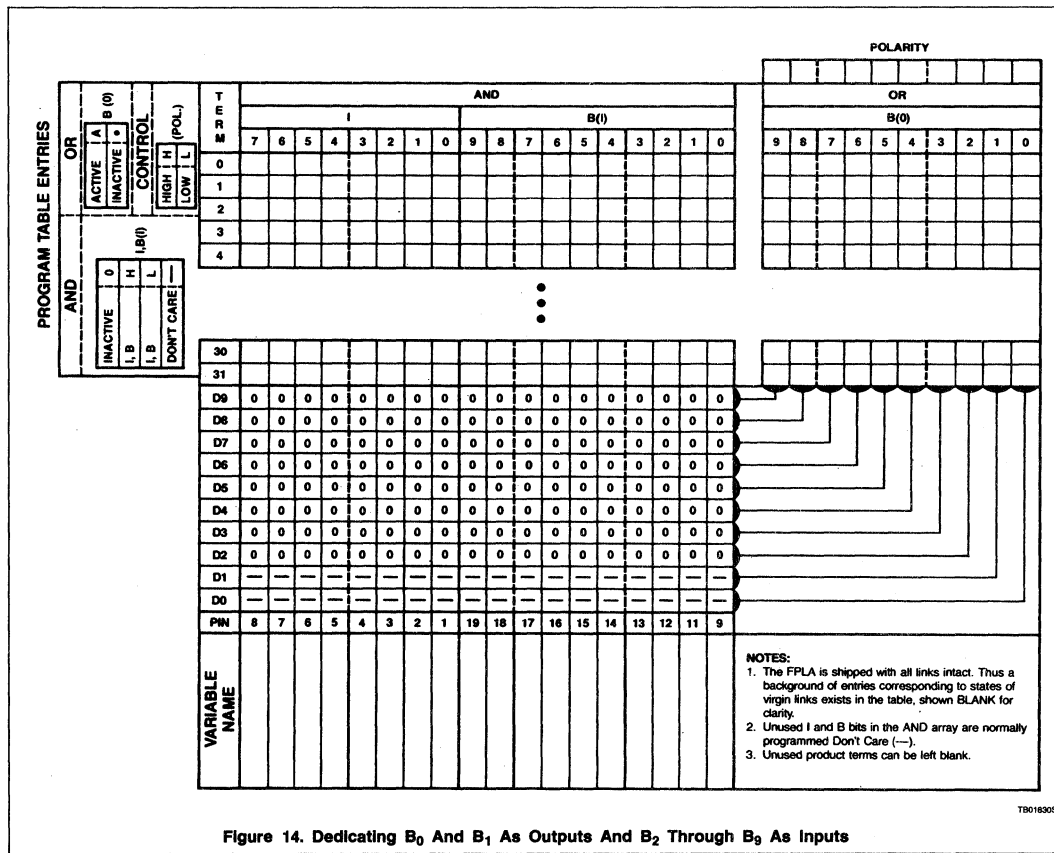


Figure 14. Dedicating B<sub>0</sub> And B<sub>1</sub> As Outputs And B<sub>2</sub> Through B<sub>9</sub> As Inputs

T8018305

# Signetics Programmable Logic

# Introduction

By placing a (—) Don't Care in each input box you are specifying that the True and Complement fuses are programmed on each Control Gate, thus permanently dedicating the B<sub>0</sub> and B<sub>1</sub> pins as outputs. By placing a (0) in all input boxes for B<sub>2</sub>–B<sub>9</sub>, you are specifying that both True and Complement fuses are intact. This causes a low (0) to be forced on half of the Control Gate inputs, guaranteeing the output of the Control Gate will be low (0). When the Control Gate outputs are low (0), the Tri-state buffer is inactive

and the B<sub>2</sub>–B<sub>9</sub> pins are enabled as inputs. All B<sub>x</sub> pin directions can be controlled in this manner.

### ACTIVE DIRECTION CONTROL

Sometimes it is necessary to be able to actively change the direction of the B<sub>x</sub> pins without permanently dedicating them. Some applications which require this include Tri-state bus enable, multi-function decoding, etc. This can easily be done by programming

the Control Gate to respond to one or more input pins. It is only necessary to select which I<sub>x</sub> and B<sub>x</sub> pins will control the pin directions and the active level HIGH (H) or LOW (L) that will be used. The PLS153 Program Table in Figure 15 shows the method of controlling B<sub>0</sub>–B<sub>9</sub> with I<sub>7</sub>. When I<sub>7</sub> is LOW (L), pins B<sub>0</sub>–B<sub>9</sub> are outputs when I<sub>7</sub> is HIGH (H), pins B<sub>0</sub>–B<sub>9</sub> are inputs. Note that by programming all other I<sub>x</sub> and B<sub>x</sub> pins as DON'T CARE (—), they are permanently disconnected from control of B<sub>x</sub> pin direction.

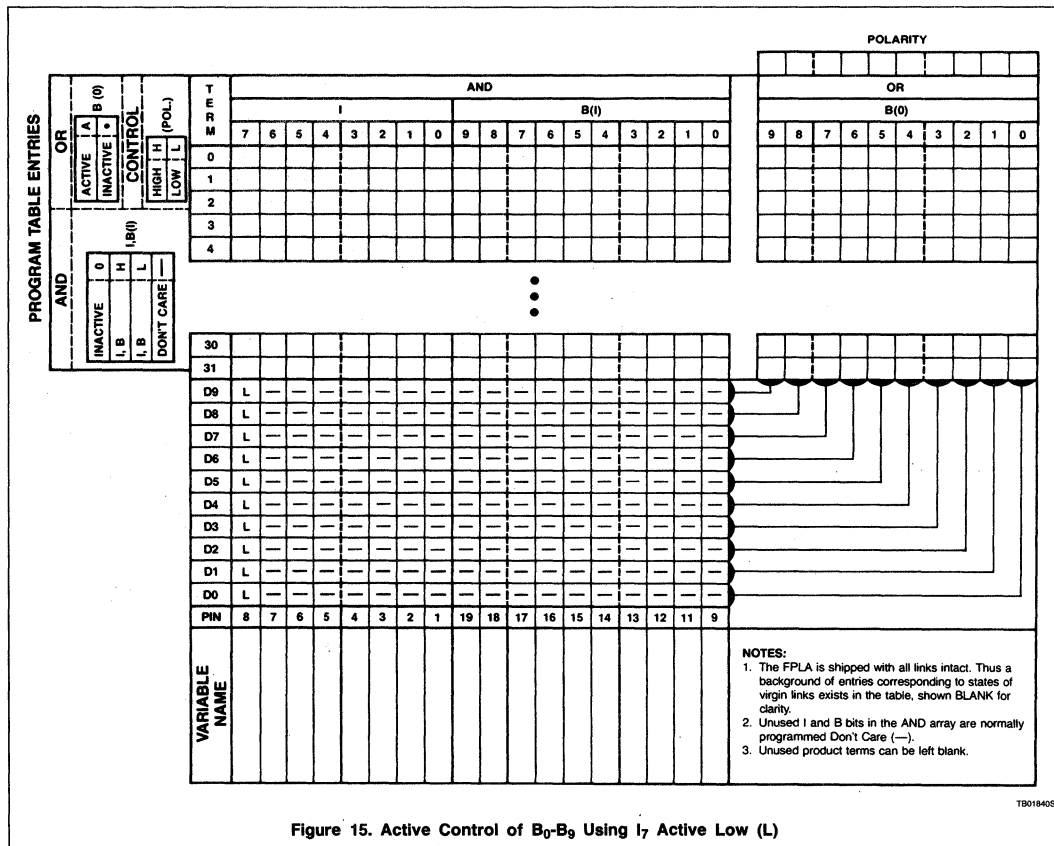


Figure 15. Active Control of B<sub>0</sub>-B<sub>9</sub> Using I<sub>7</sub> Active Low (L)

TB018405

# Signetics Programmable Logic

# Introduction

The previous 28-pin logic synthesis example (Page 2-5) could be done on the PLS153 as follows:

$$X_0 = AB + \bar{C}D + B\bar{D}$$

$$\bar{X}_1 = \bar{A}B + \bar{C}D + EFG$$

Note that B<sub>0</sub> was used as a CHANGE input. When B<sub>0</sub> is HIGH (H) the outputs appear on B<sub>8</sub> and B<sub>9</sub>. When B<sub>0</sub> is LOW (L), the outputs appear on B<sub>6</sub> and B<sub>7</sub>. B<sub>1</sub> through B<sub>5</sub> are not used and therefore left unprogrammed.

## THE ROLE OF AMAZE (AUTOMATIC MAP AND ZAP ENTRY) IN LOGIC DEVELOPMENT

AMAZE is a software development package which provides the logic designer with a multi-mode data input capability. This software package, which has been developed for both mainframe and microcomputer environments, allows data entry in the program table format previously described, or on one of two equation formats. Both Boolean Logic Equations and State Variable Equations are supported on AMAZE. In addition to the flexible input formats offered, AMAZE also provides full

device documentation, and a functional simulator. The simulator features manual and automatic modes. In the automatic mode, the software generates a set of input vectors which can be used as functional test vectors. The manual mode is an interactive procedure which allows the designer to input vectors to the simulator. The AMAZE simulator then responds with the appropriate output vector. In both modes, the simulator uses the functional model generated by the user's input data.

For more information on AMAZE refer to Section 8 of the 1987 Signetics Programmable Logic Data Manual.

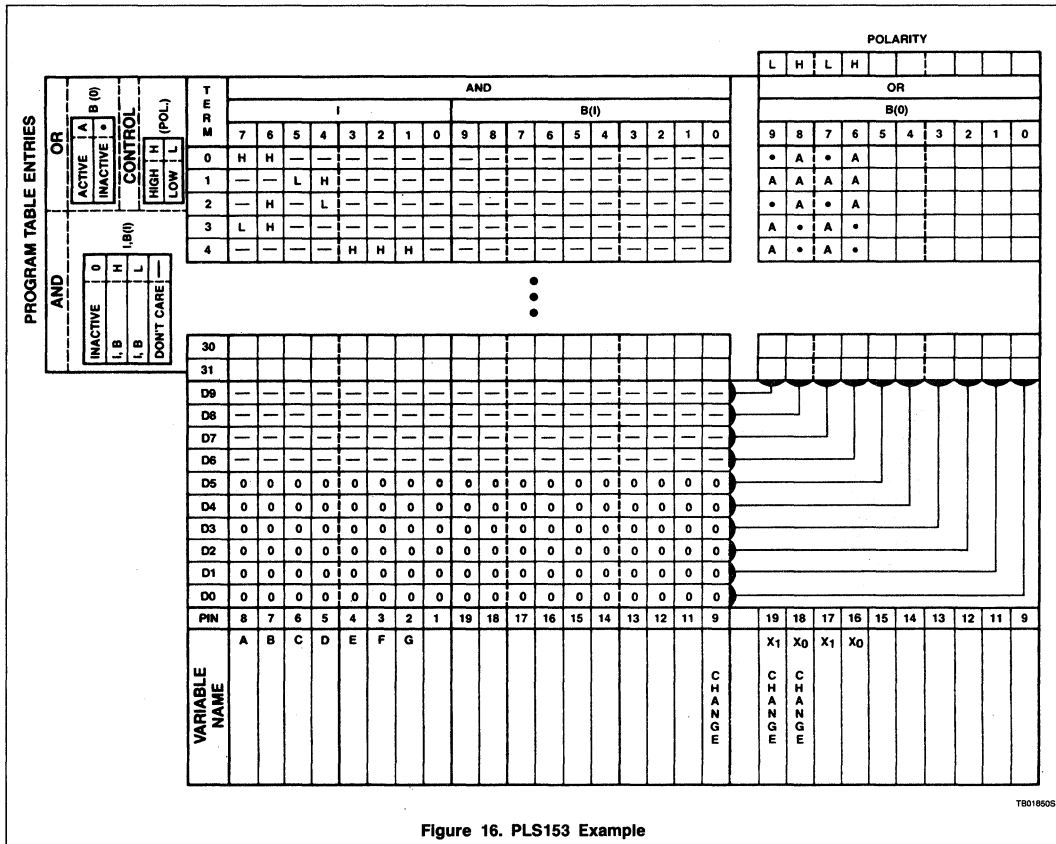


Figure 16. PLS153 Example



# Signetics Programmable Logic

# Introduction

## SEQUENTIAL LOGIC CONSIDERATIONS

The PLS105 and PLS155 - PLS159 represent significant increases in complexity when compared to the combinatorial logic devices pre-

viously discussed. By combining the AND/OR combinatorial logic with clocked output flip-flops and appropriate feedback, Signetics has created the first family of totally flexible sequential logic machines.

The PLS105 FPLS (Field-Programmable Logic Sequencer) is an example of a high-order machine whose applications are manifold. Application areas for this device include high-speed data controllers, microprocessor and minicomputer bus arbitration, industrial controls, timing generation, multi-function counters and shift registers, and microprocessor-driven microcontrollers. The PLS105 is fully capable of performing fast sequential operations in relatively low-speed processor systems. By placing repetitive sequential operations on the PLS105, processor overhead is reduced. Each PLS105 can be viewed as a high-speed, 48-state subroutine.

The following pages summarize the PLS105 architecture and features.

### FPLS Architecture

The PLS105 Logic Sequencer is a programmable state machine of the Mealy type, in which the output is a function of the present state and the present input.

With the FPLS a user can program any logic sequence expressed as a series of jumps between stable states, triggered by a valid input condition (I) at clock time (t). All stable states are arbitrarily assigned and stored in the State Register. The logic output of the machine is also programmable, and is stored in the Output Register.

### Clocked Sequence

A synchronous logic sequence can be represented as a group of circles interconnected with arrows. The circles represent stable states, labeled with an arbitrary numerical code (binary, hex, etc.) corresponding to discrete states of a suitable register. The arrows represent state transitions, labeled with symbols denoting the jump condition and the required change in output. The number of states in the sequence depends on the length and complexity of the desired algorithm.

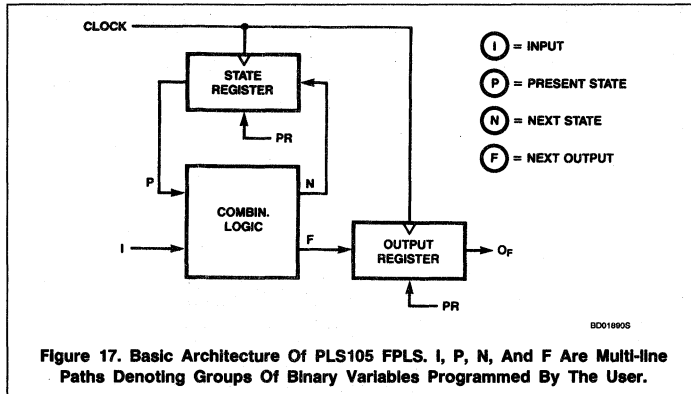


Figure 17. Basic Architecture Of PLS105 FPLS. I, P, N, And F Are Multi-line Paths Denoting Groups Of Binary Variables Programmed By The User.

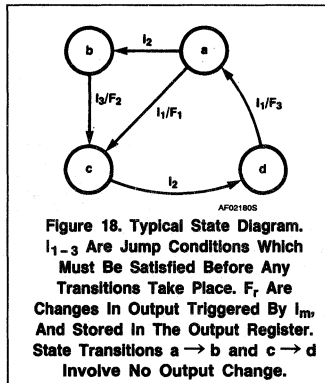


Figure 18. Typical State Diagram. I<sub>1-3</sub> Are Jump Conditions Which Must Be Satisfied Before Any Transitions Take Place. F<sub>r</sub> Are Changes In Output Triggered By I<sub>m</sub> And Stored In The Output Register. State Transitions a → b and c → d Involve No Output Change.

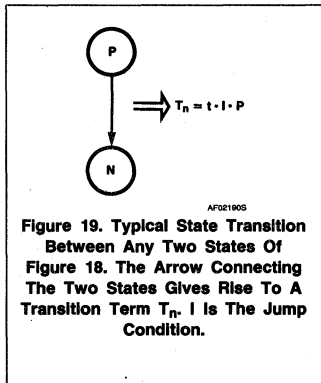


Figure 19. Typical State Transition Between Any Two States Of Figure 18. The Arrow Connecting The Two States Gives Rise To A Transition Term T<sub>n</sub>. I Is The Jump Condition.

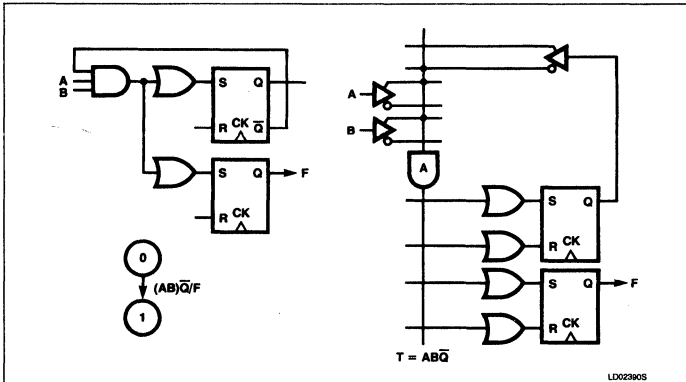


Figure 20. Typical State Jump From State (0) To State (1), If Inputs A = B = "1". The Jump Also Forces F = "1", As Required.

**State Jumps**

The state from which a jump originates is referred to as the Present state (P), and the state to which a jump terminates is defined as the Next state (N). A state jump always causes a change in state, but may or may not cause a change in machine output (F).

State jumps can occur only via "transition terms"  $T_n$ . These are logical AND functions of the clock (t), the Present state (P), and a valid input (I). Since the clock is actually applied to the State Register,  $T_n = I \cdot P$ . When  $T_n$  is "true", a control signal is generated and used at clock time (t) to force the contents of the State Register from (P) to (N), and to change the contents of the Output Register (if necessary). The simple state jump in Figure 20, involving 2 inputs, 1 state bit, and 1 output bit, illustrates the equivalence of discrete and programmable logic implementations.

**FPLS Logic Structure**

The FPLS consists of programmable AND and OR gate arrays which control the Set and Reset inputs of a State Register, as well as monitor its output via an internal feedback path. The arrays also control an independent Output Register, added to store output commands generated during state transitions, and to hold the output constant during state sequences involving no output changes. If desired, any number of bits of the Output Register can be used to extend the width of the State Register, via external feedback.

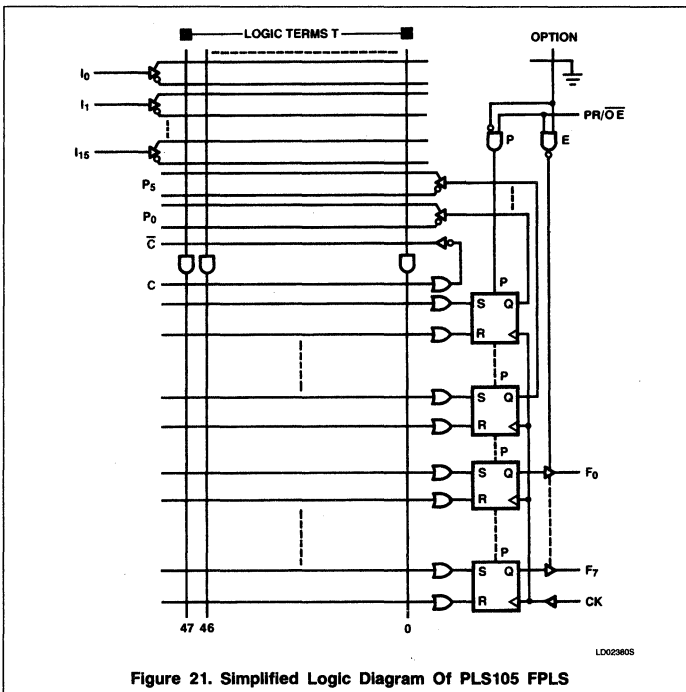


Figure 21. Simplified Logic Diagram Of PLS105 FPLS

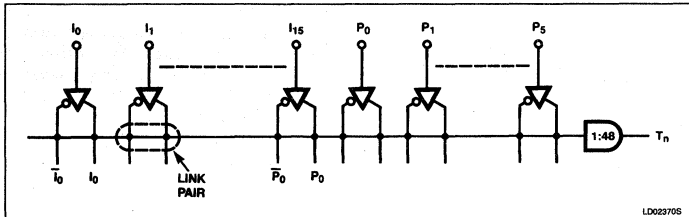


Figure 22. Typical AND Gate Coupled To (I) And (P) Inputs. If At Least One Link Pair Remains Intact,  $T_n$  Is Unconditionally Forced Low.

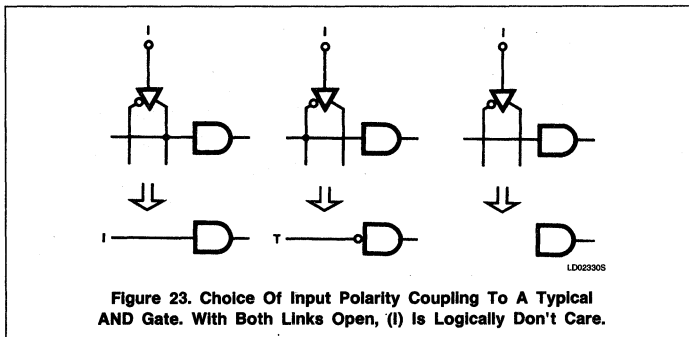


Figure 23. Choice Of Input Polarity Coupling To A Typical AND Gate. With Both Links Open, (I) Is Logically Don't Care.

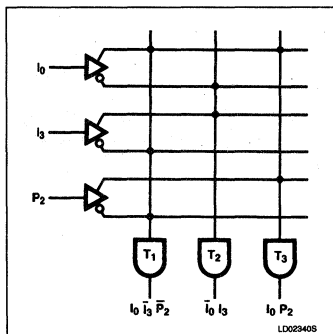


Figure 24. Typical Transition Terms Involving Arbitrary Inputs And State Variables. All Remaining Gate Inputs Are Programmed Don't Care. Note That  $T_2$  Output Is State Independent.

**Input Buffers**

16 external inputs ( $I_m$ ) and 6 internal inputs ( $P_n$ ), fed back from the State Register, are combined in the AND array through two sets of True/Complement (T/C) buffers. There are a total of 22 T/C buffers, all connected to multi-input AND gates via fusible links which are initially intact.

Selective fusing of these links allows coupling either True, Complement, or Don't Care values of ( $I_m$ ) and ( $P_n$ ).

**"AND" Array**

State jumps and output changes are triggered at clock time by valid transition terms  $T_n$ . These are logical AND functions of the present state (P) and the present input (I).

The FPLS AND Array contains a total of 48 AND gates. Each gate has 45 inputs — 44 connected to 22 T/C input buffers, and 1 dedicated to the Complement Array. The outputs of all AND gates are propagated through the OR Array, and used at clock time (t) to force the contents of the State Register from (P) to (N). They are also used to control the Output Register, so that the FPLS 8-bit output  $F_r$  is a function of the inputs and present state.

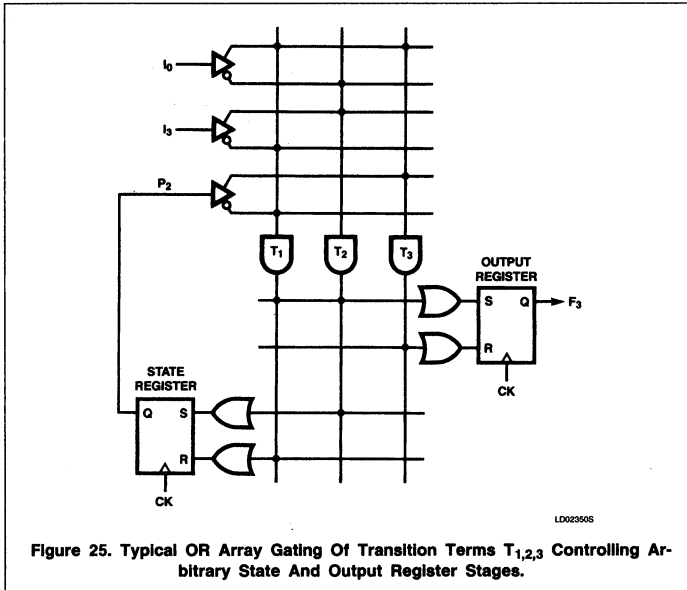


Figure 25. Typical OR Array Gating Of Transition Terms  $T_{1,2,3}$  Controlling Arbitrary State And Output Register Stages.

**"OR" Array**

In general, a clocked sequence will consist of several stable states and transitions, as determined by the complexity of the desired algorithm. All state and output changes in the state diagram imply changes in the contents of State and Output Registers.

Thus, each flip-flop in both registers may need to be conditionally set or reset several times with  $T_n$  commands. This is accomplished by selectively ORing through a programmable OR Array all AND gate outputs  $T_n$  necessary to activate the proper flip-flop control inputs.

The FPLS OR Array consists of 14 pairs of OR gates, controlling the S/R inputs of 14 State and Output Register stages, and a single OR gate for the Complement Array. All gates have 48 inputs for connecting to all 48 AND gates.

**Complement Array**

The COMPLEMENT Array provides an asynchronous feedback path from the OR Array back to the AND Array.

This structure enables the FPLS to perform both direct and complement sequential state jumps with a minimum of transition (AND) terms.

Typically direct jumps, such as  $T_1$  and  $T_2$  in Figure 27 require only a single AND gate each.

But a complement jump such as  $T_3$  generally requires many AND gates if implemented as a direct jump. However, by using the Complement Array, the logic requirements for this type of jump can be handled with just one more gate from the AND Array.

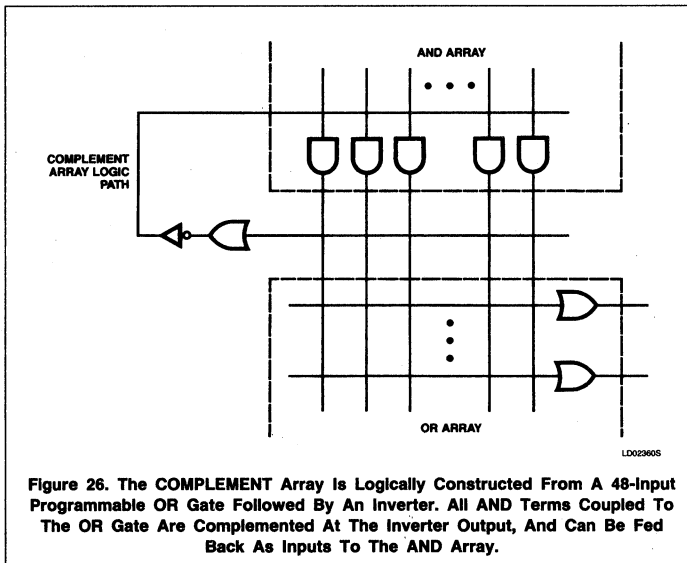
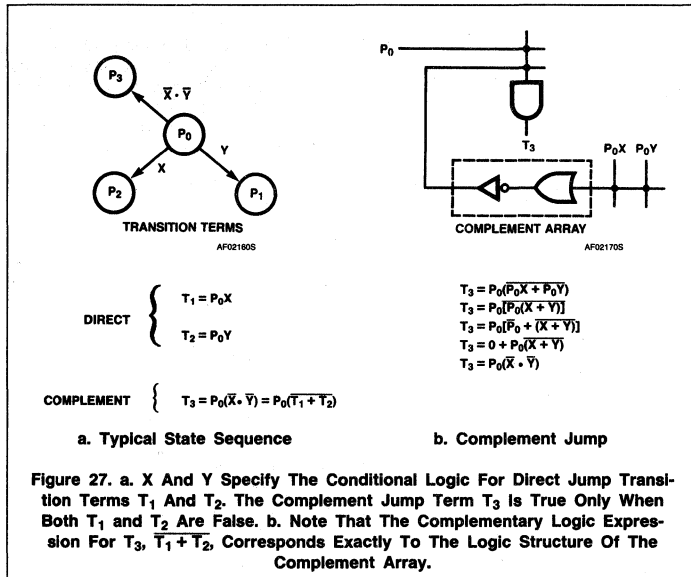
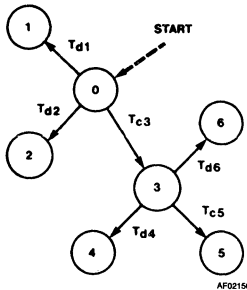


Figure 26. The COMPLEMENT Array is Logically Constructed From A 48-Input Programmable OR Gate Followed By An Inverter. All AND Terms Coupled To The OR Gate Are Complementated At The Inverter Output, And Can Be Fed Back As Inputs To The AND Array.



As indicated in Figure 28, the single Complement Array gate may be used for many states of the state diagram. This happens because all transition terms linked to the OR gate include the present state as a part of their conditional logic. In any particular state, only those transition terms which are a function of that state are enabled; all other terms coupled to different states are disabled and do not influence the output of the Complement Array. As a general rule of thumb, the Complement Array can be used as many times as there are states.



a. State Diagram

$$T_{d1} = i_0 i_1 P_0$$

$$T_{d2} = i_2 P_0$$

$$T_{c3} = \overline{(T_{d1} + T_{d2})} P_0 = \overline{(i_0 i_1 + i_2)} P_0$$

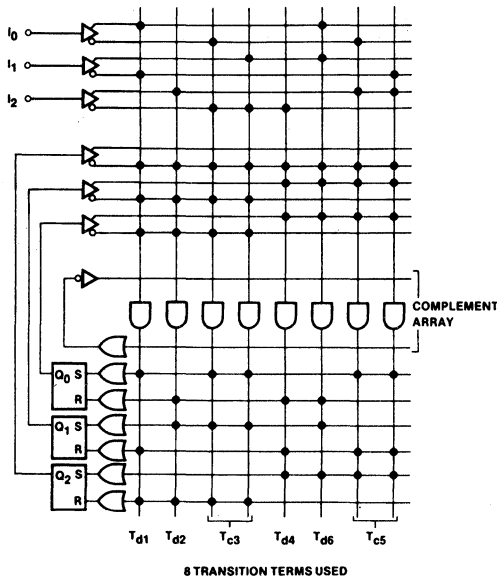
$$T_{d4} = i_2 P_3$$

$$T_{d6} = i_0 i_1 P_3$$

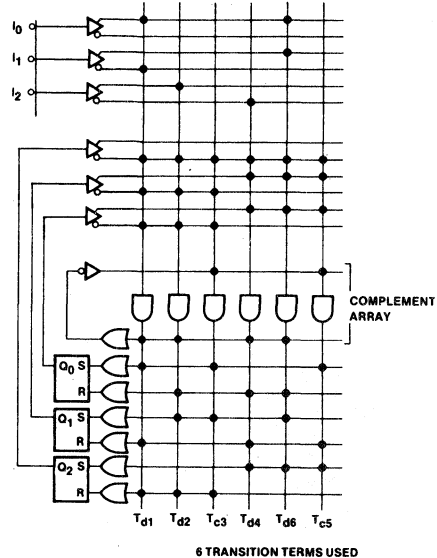
$$T_{c5} = \overline{(T_{d4} + T_{d6})} P_3 = \overline{(i_0 i_1 + i_2)} P_3$$

$T_{cn}$  = COMPLEMENT STATE TRANSITION TERM  
 $T_{dn}$  = DIRECT STATE TRANSITION TERM  
 $P_n$  = PRESENT STATE

b. Logic Definition



c. State Logic Without Using The Complement Array



d. State Logic Using The Complement Array

Figure 28. Logic Reduction With The Complement Array. The Logic State Diagram In (a) Includes Complement Jumps  $T_{C3}$  And  $T_{C5}$  Defined In (b). When Using The Complement Array, A Savings Of 2 Transition Terms Results, As Shown In (c) And (d).

# Signetics Programmable Logic

# Introduction

**Table 5. 20-Pin PLD Sequential Family Summary**

DEVICE	REGISTERED OUTPUTS	OUTPUT TYPE
PLS155	4-Bit Register	T.S.
PLS157	6-Bit Register	T.S.
PLS159	8-Bit Register	T.S.

### PLD-20 SEQUENTIAL DEVICES

The 20-pin PLD family also includes sequential devices. These devices are all similar in architecture. The major difference consists of the number of outputs which are registered.

### FEATURES

The PLD-20 sequencers have been designed with a maximum of flexibility in mind. Each element of the architecture contributes to the ease of use the PLD-20 family provides. Each part has the features listed:

- J/K flip-flop Output Register
- Dynamic control of flip-flop type (J/K or D)

- Parallel bus load
- Asynchronous preset and reset capability
- Combinational I/O pins
- Programmable enable pins
- Output feedback to AND Array available

### OUTPUT REGISTER

The Output Register of the PLD-20 sequential devices are comprised of fully implemented J/K flip-flops. Each flip-flop is positive edge-triggered from a common clock.

In addition, a dynamically controllable "fold-back" buffer between the J and K inputs to the flip-flop allows the designer to change the function of the flip-flop from J/K to D-type under the control of the flip-flop control term (F). A fuse allows the designer to dedicate the flip-flop as a permanent D-type by programming the fuse.

By leaving the fuse intact, the flip-flop control line (F) is maintained as the active mode control. If the output of F is logic Low (0), then the flip-flop is configured as a D-type. If the output of F is a logic High (1), then the flip-flop is configured as a J/K type. Term F is programmed in the same manner as the 32 logic transition terms (T). It is well to note at this time that when the flip-flop is configured as a D-type, the OR-term driving the K input of the flip-flop must not be active.

During all modes of operation, the output register data is fed back to the AND Array. This feedback is used to establish present-state to next-state jumps.

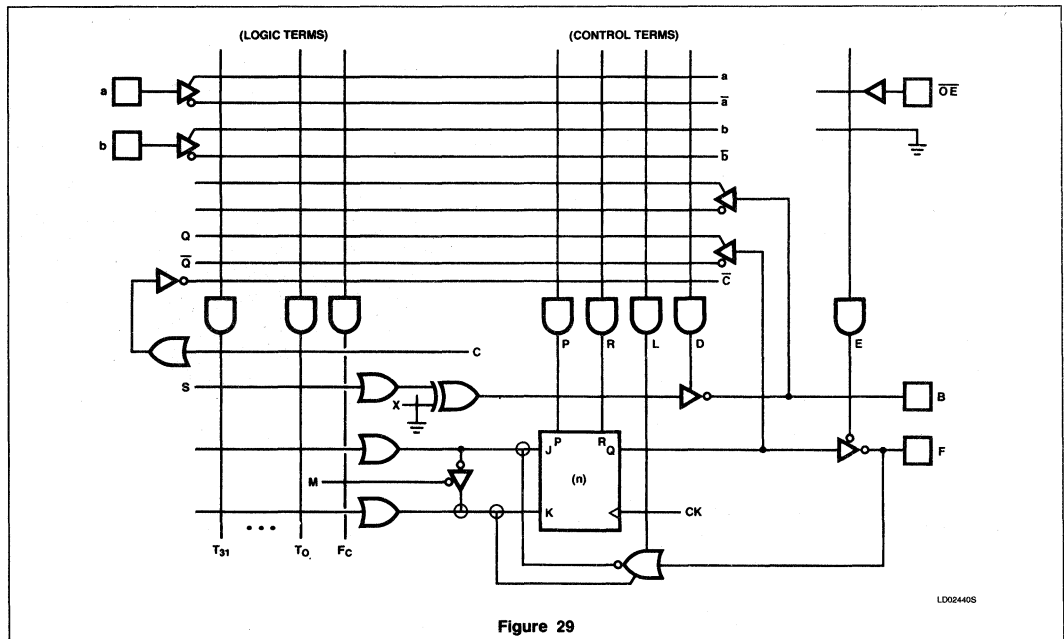


Figure 29

LD024405

**PARALLEL BUS LOADING**

The Output Register may be loaded from the bus via the load control term (L). This feature

forces the data contained on the  $F_0$  pin into the J/K flip-flop. It operates synchronously with the rising edge of the clock. This feature

can be used to preload a state into the machine, or to latch input data into the AND Array.

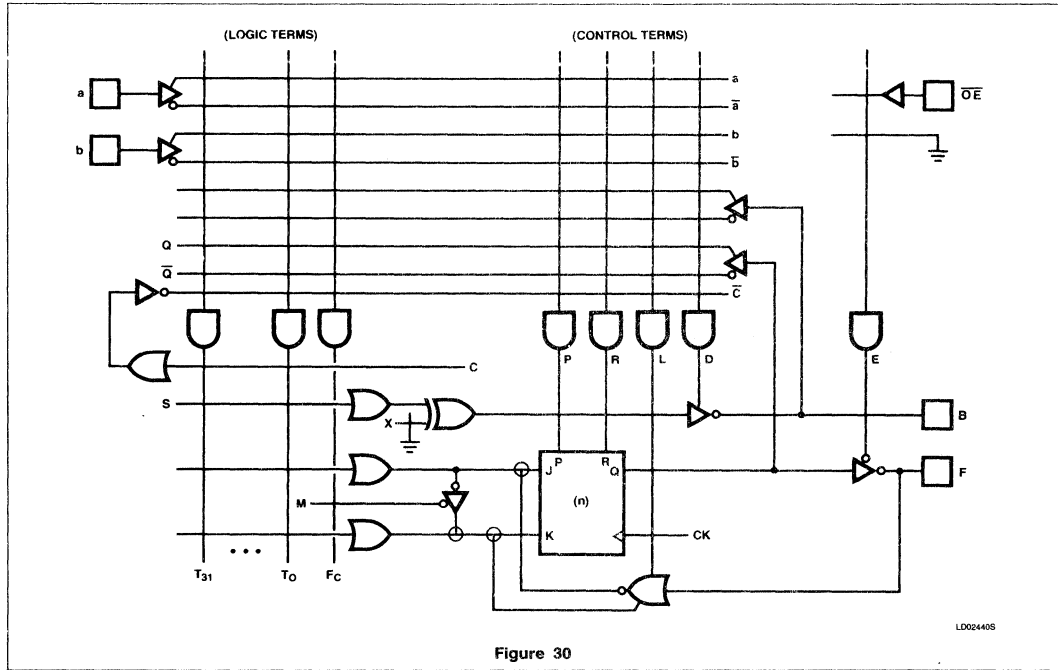


Figure 30

LD024405



# Signetics Programmable Logic

# Introduction

## PRESET AND RESET

Asynchronous preset and reset capability has been provided on the PLD-20 sequential

devices. This feature is controlled in the AND-Array on the PLS159. However, the preset and reset function on the PLS155 is controlled via the OR Array. The PLS157 preset

and reset controls are split between the two arrays. See the individual data sheets for details.

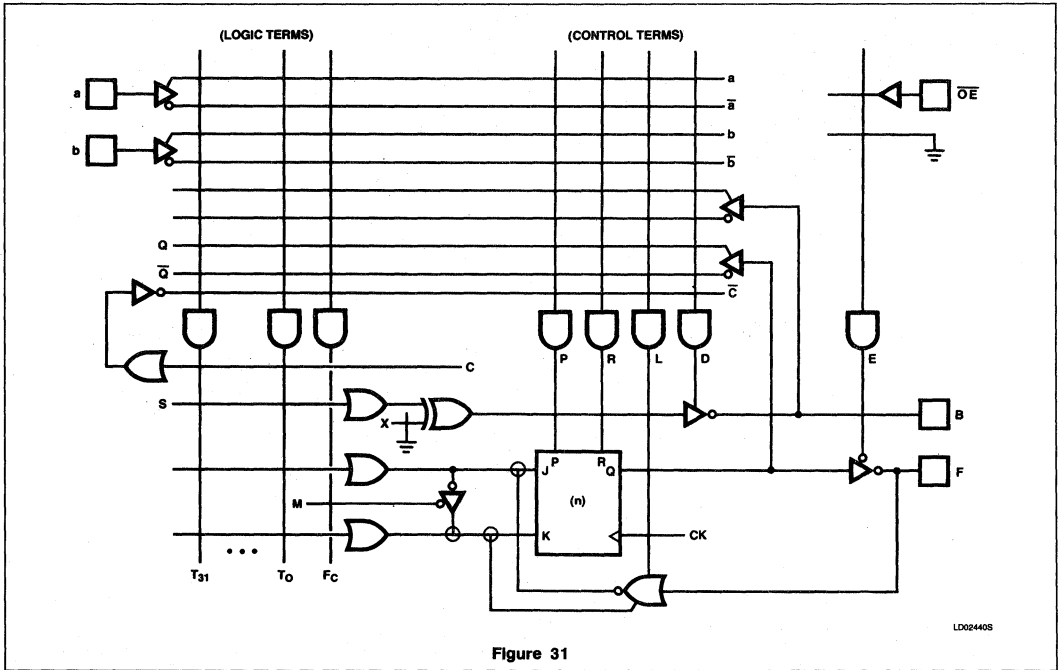


Figure 31

**COMBINATIONAL I/O PINS**

Each PLD-20 sequential device has a number of combinational pins ( $B_n$ ) which can be

programmed and used exactly as the B pins on the PLS153 FPLA. The direction control term (D) establishes the data flow on these

pins. The individual data sheet should be consulted for the quantity and pin number assignment for each device.

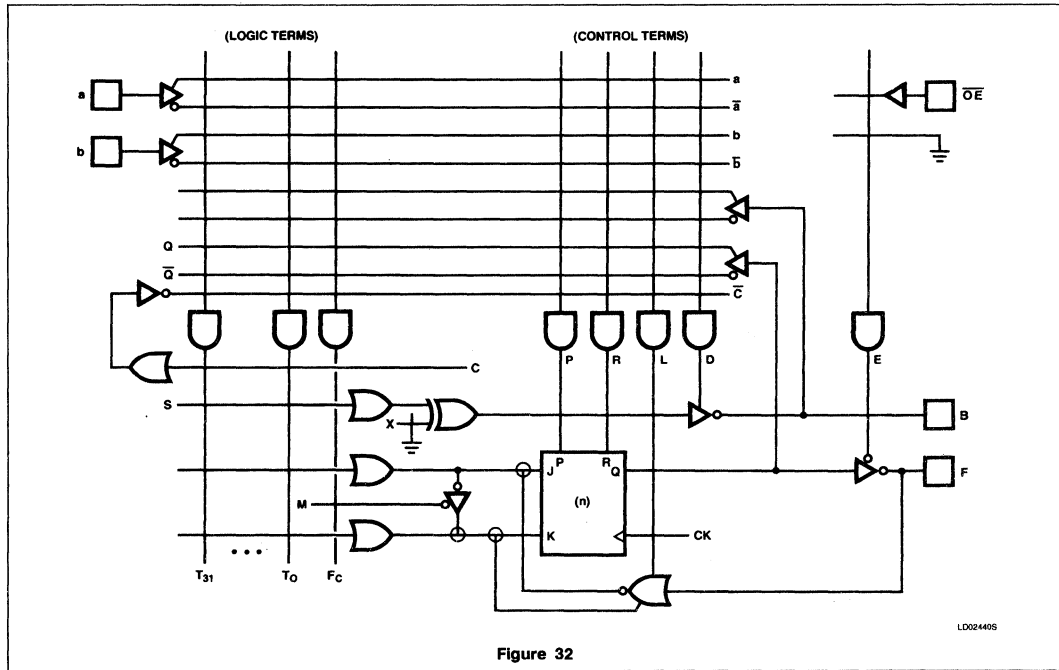


Figure 32

# Signetics Programmable Logic

# Introduction

## PLA vs. PAL<sup>®</sup> ARRAYS

The PLA architecture provides the most efficient means of implementing logic. The 28-pin devices, PLS100/101 and PLS103 are unique in their ability to directly implement logic.

They offer the most useable P-terms, the highest number of inputs and outputs, and the most straightforward programming of any logic device in the industry. The following diagrams illustrate the relative ease of programming the flexible PLA structure against the fixed OR Array of the PAL.

The Signetics approach to programmable logic gives the designer the flexibility he needs where he needs it—in the device itself. The combination of totally flexible architecture, more useable product terms, and ease of logic implementation make Signetics Programmable Logic Family the obvious choice.

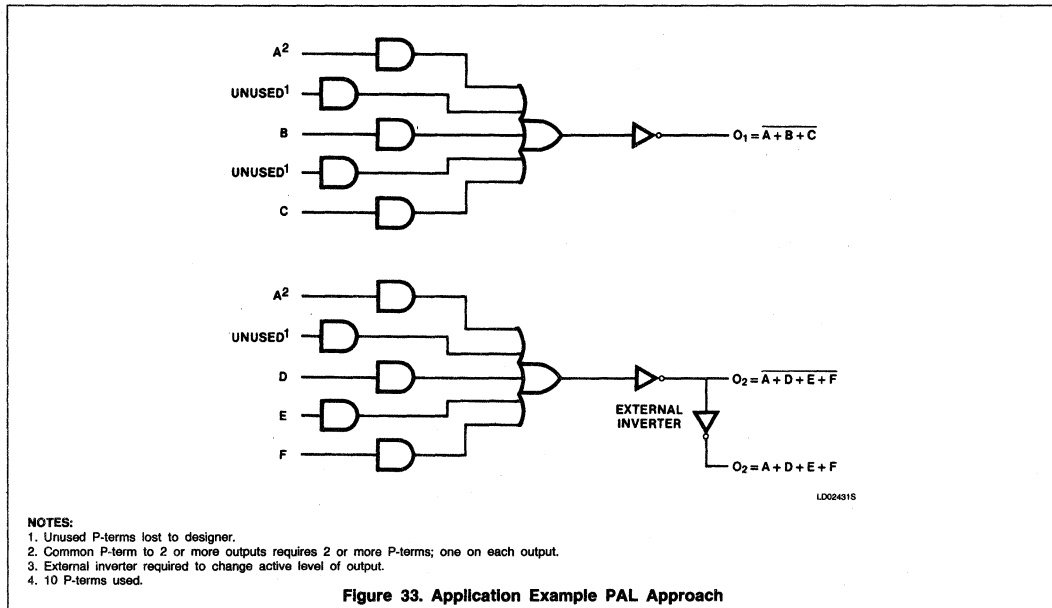


Figure 33. Application Example PAL Approach

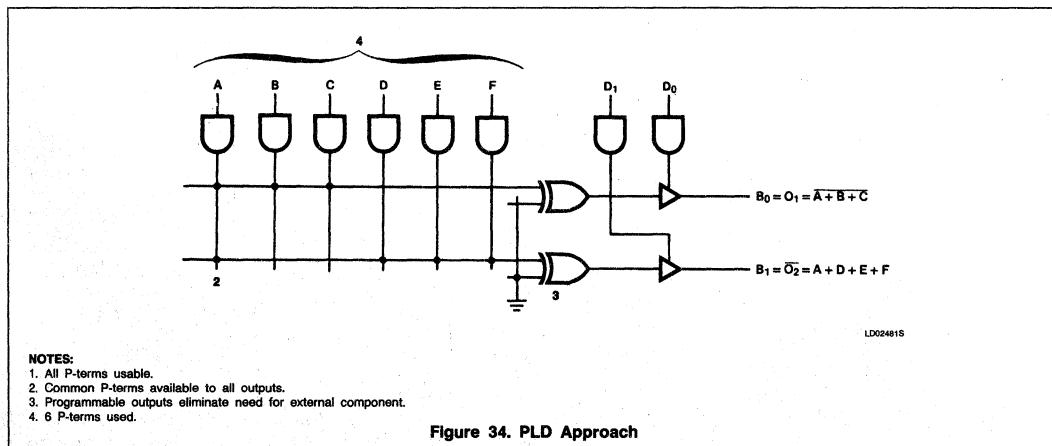


Figure 34. PLD Approach

\*PAL is a registered trademark of Monolithic Memories, Inc.

## Signetics Programmable Logic

## Introduction

FPLA PLS153/153A COMPATIBLE WITH PAL<sup>®</sup> LOW/MEDIUM COMPLEXITY PRODUCT, i.e., 14H/L4 16H/L2 16LS 10H/LS 16C1, 12H/LS

## ADDITIONAL FEATURES:

1. PROGRAMMABLE OUTPUT POLARITY
2. PROGRAMMABLE OR ARRAY
3. 10 OUTPUTS vs. 8 FOR PAL
4. 32 PRODUCT TERMS vs. 16 FOR LOW AND MEDIUM PALs

FPLS PLS155/157/159 COMPATIBLE WITH PAL REGISTERED TYPES 16R3, 16R5, 16R4, etc.

## ADDITIONAL FEATURES:

1. PROGRAMMABLE AND/OR, ALSO CONTROLLABLE FLIP FLOPS; AS D, TOGGLE, OR JK

\*PAL is a trademark of Monolithic Memories, Inc.

Figure 35. PLD-20 Product Comparison

## PLD FAMILY SUMMARY

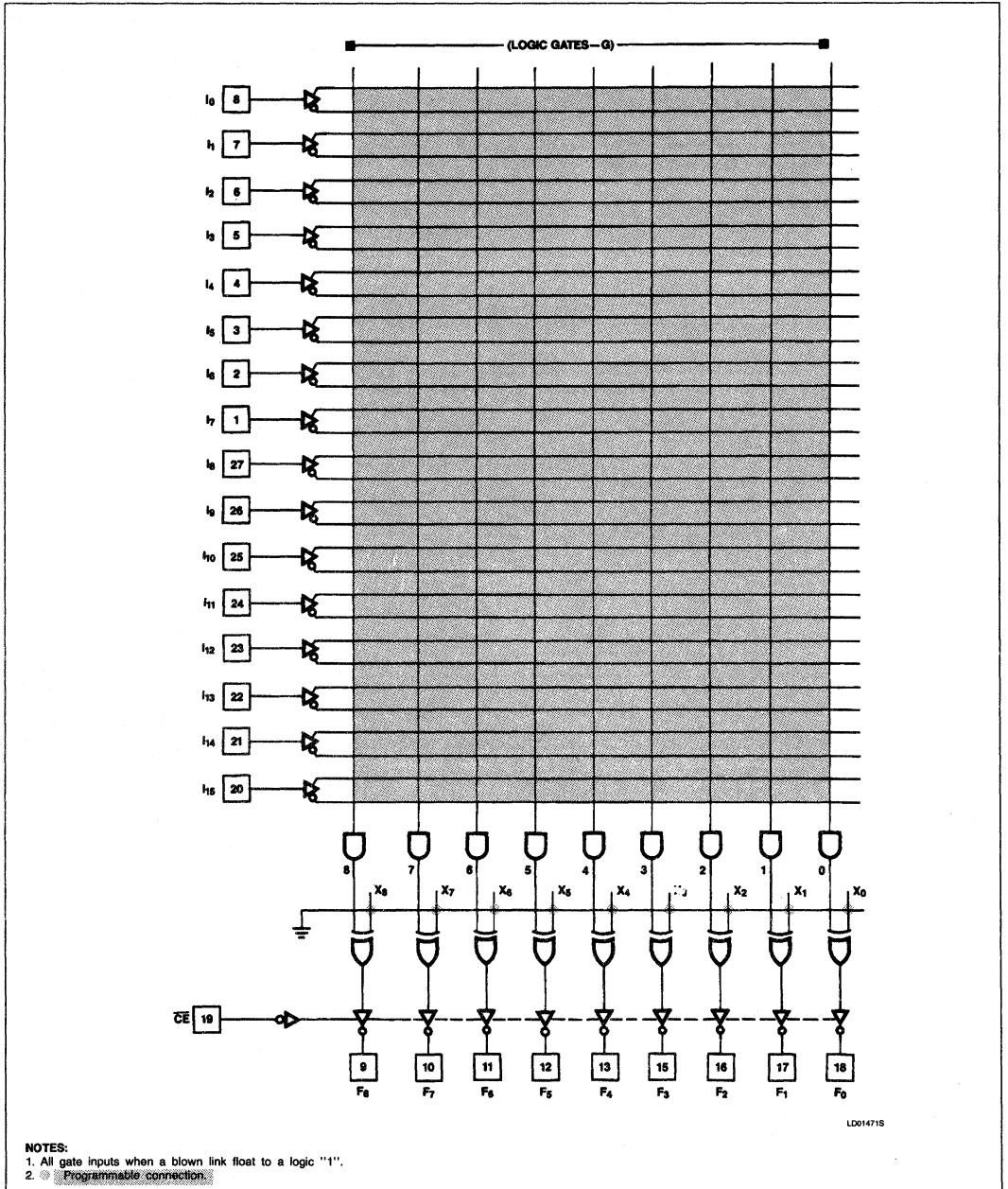
Field-Programmable Gate Arrays			
PLS103	16 × 9	28-Pin	Tri-state (Combinatorial)
PLS151	18 × 12	20-Pin	Tri-state (Combinatorial)
Field-Programmable Address Decoders			
PLS162	16 × 5	24-Pin	Tri-state (Combinatorial)
PLS163	12 × 9	24-Pin	Tri-state (Combinatorial)
Field-Programmable Logic Arrays			
PLS100	16 × 48 × 8	28-Pin	Tri-state (Combinatorial)
PLS101	16 × 48 × 8	28-Pin	Open Collector (Combinatorial)
PLS153/153A/PLHS153	18 × 45 × 10	20-Pin	Tri-state (Combinatorial)
PLS161	12 × 48 × 8	24-Pin	Tri-state (Combinatorial)
PLS173/PLUS173	22 × 45 × 10	24-Pin	Tri-state (Combinatorial)
PLHS473	20 × 24 × 11	24-Pin	Tri-state and Open Collector
PLC473	20 × 24 × 11	24-Pin	Tri-state and Open Collector (CMOS)
PLHS18P8A	18 × 72 × 8	20-Pin	Tri-state (Combinatorial); Fixed OR Array
Field-Programmable Logic Sequencers			
PLS105/105A	16 × 48 × 8	28-Pin	Tri-state Mealy State Machine
PLS155	16 × 42 × 12	20-Pin	Tri-state 4-Bit Sequencer
PLS157	16 × 42 × 12	20-Pin	Tri-state 6-Bit Sequencer
PLS159/159A	16 × 42 × 12	20-Pin	Tri-state 8-Bit Sequencer
PLS167/167A	14 × 48 × 6	24-Pin	Tri-state Mealy State Machine
PLS168/168A	12 × 48 × 8	24-Pin	Tri-state Mealy State Machine
PLS179	20 × 42 × 8	24-Pin	Tri-state 8-Bit Sequencer
PLUS405A	16 × 64 × 8	28-Pin	Tri-state Mealy State Machine
Field-Programmable Macro Logic			
PLHS501	32 × 72 × 24	52-Pin	Random Logic Unit
PLHS502	32 × 64 × 24	52-Pin	Registered Logic Unit

# Signetics Programmable Logic

# Introduction

## FPGA LOGIC DIAGRAM

PLS103

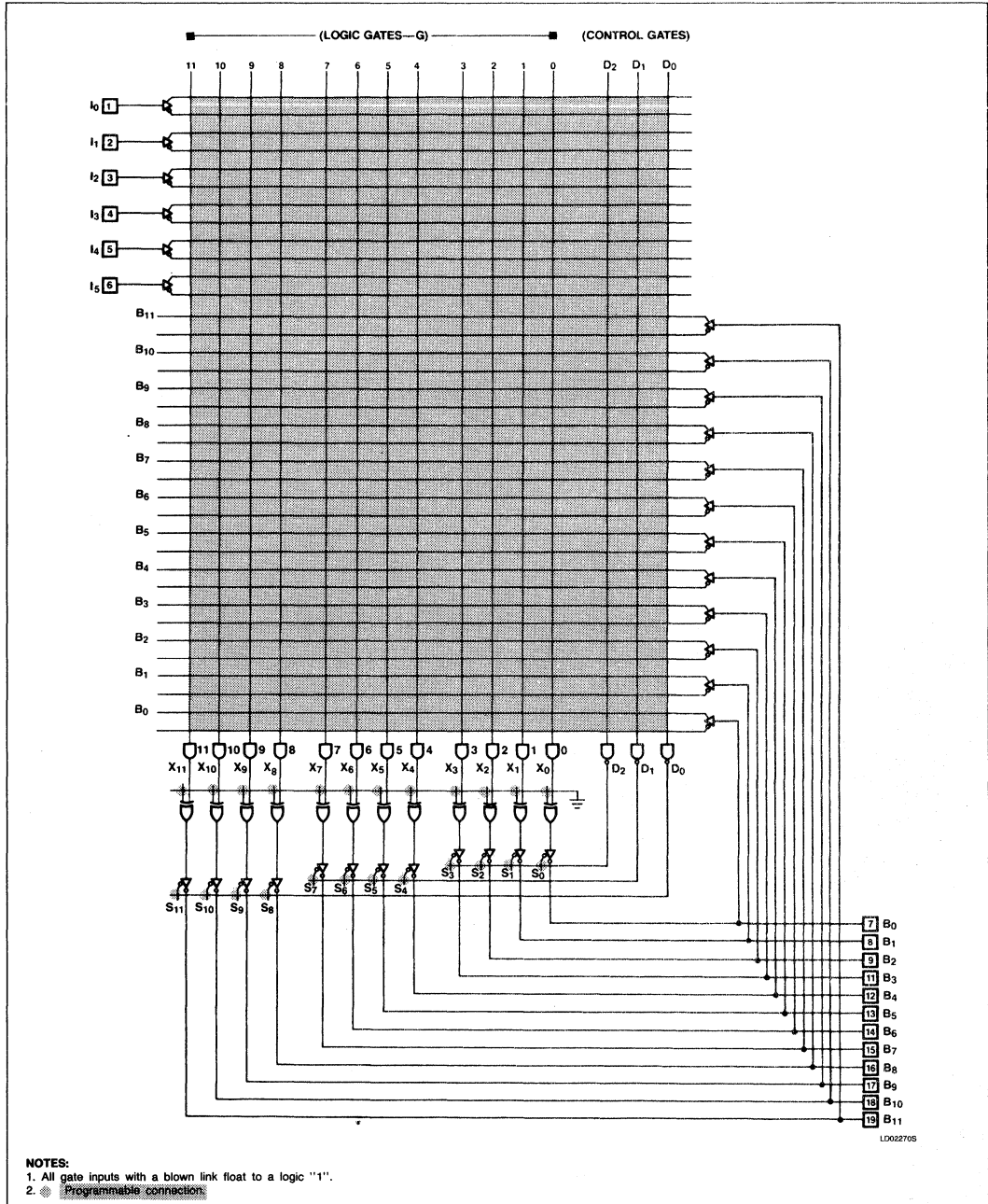


# Signetics Programmable Logic

# Introduction

## FPGA LOGIC DIAGRAM

PLS151



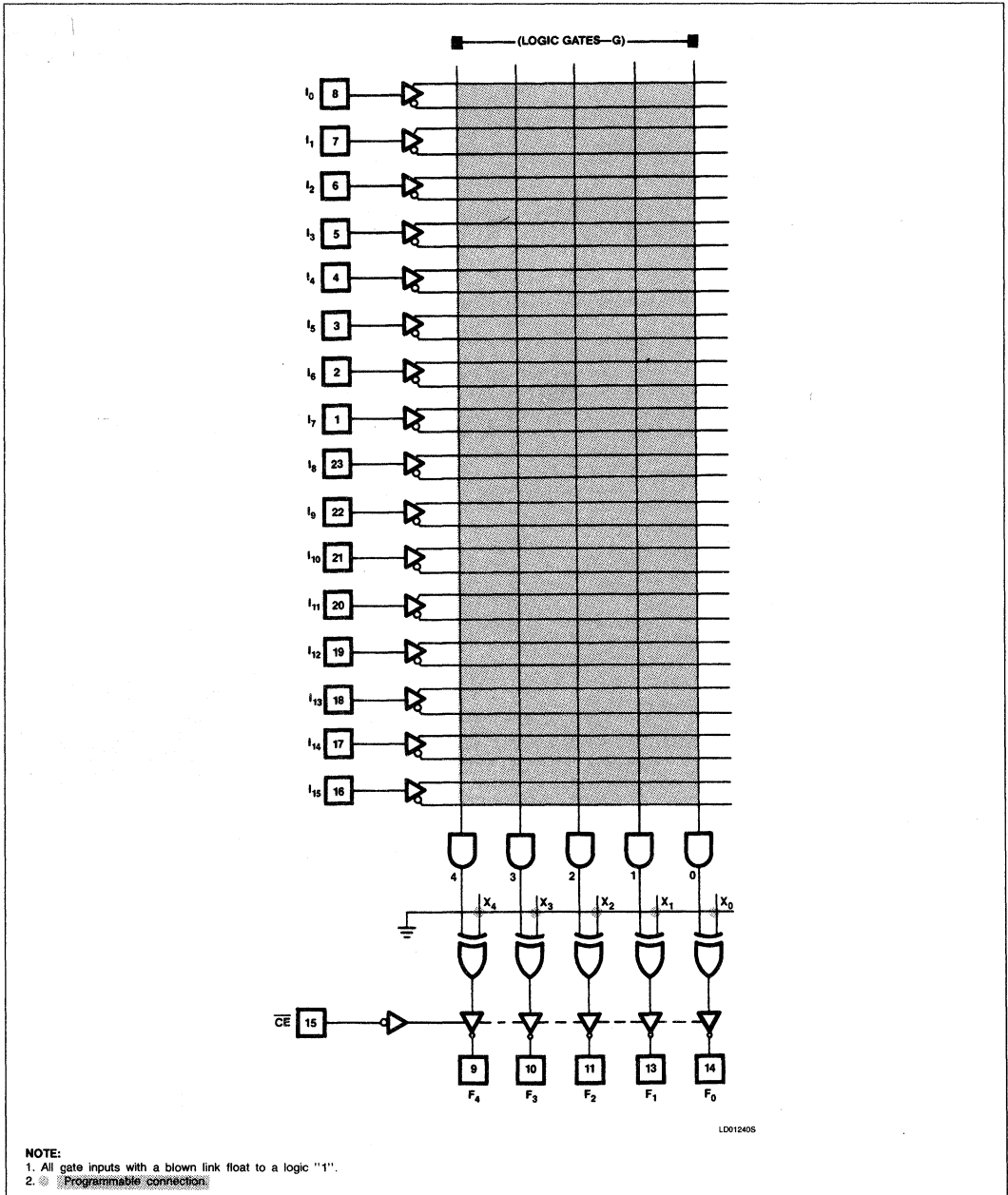
**NOTES:**

1. All gate inputs with a blown link float to a logic "1".
2. Programmable connection.

LD022705

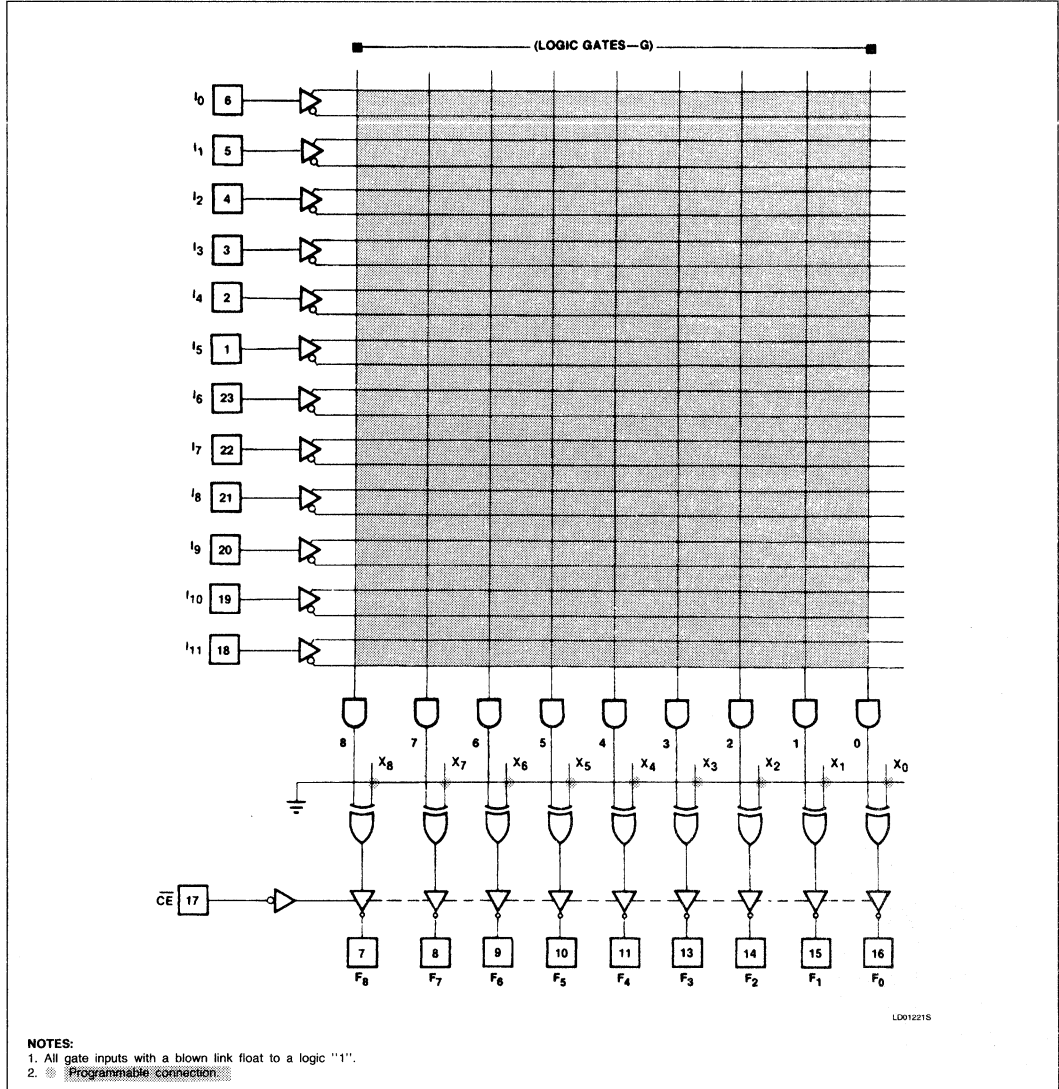
FPAD LOGIC DIAGRAM

PLS162



FPAD LOGIC DIAGRAM

PLS163



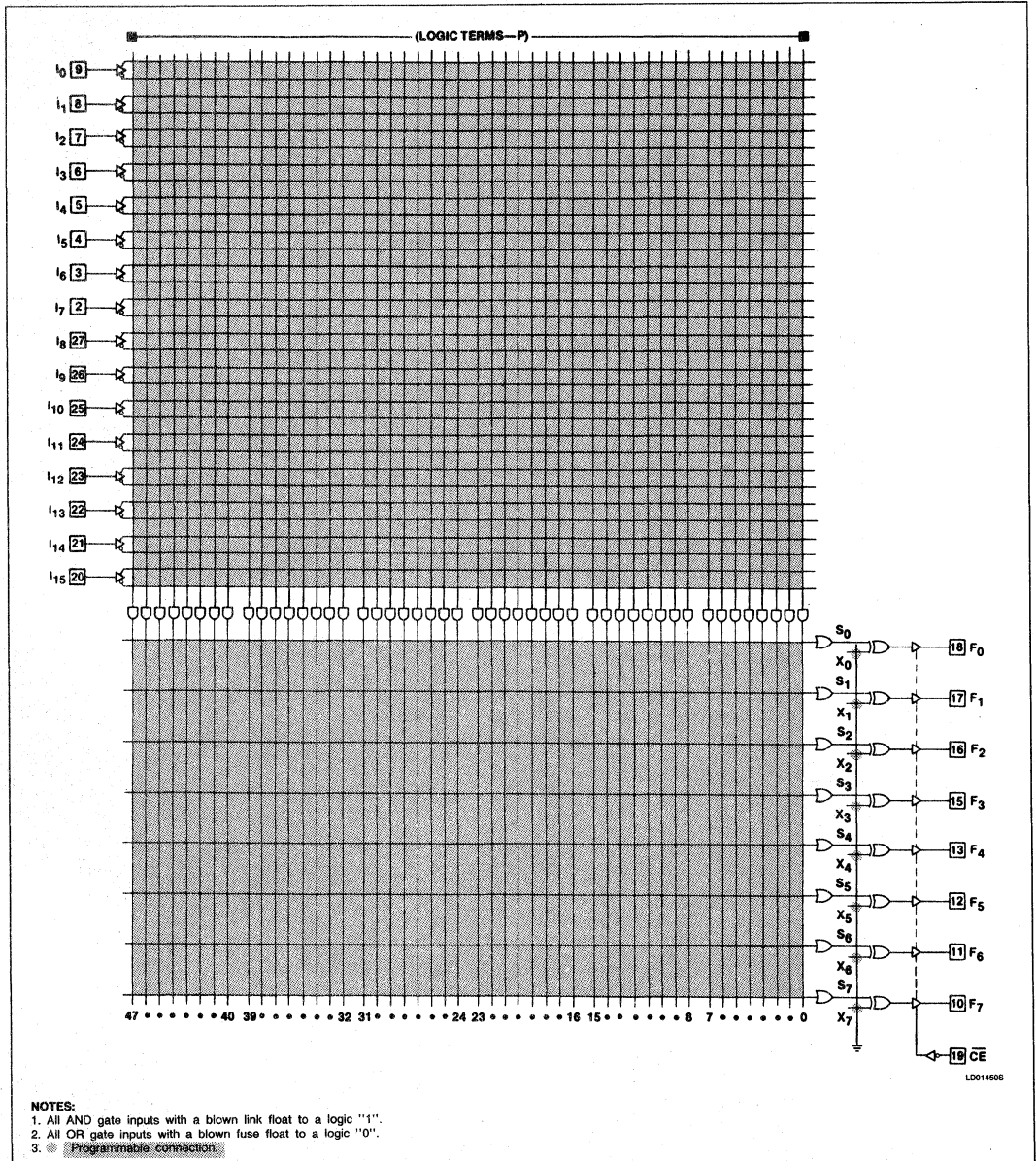


# Signetics Programmable Logic

# Introduction

## FPLA LOGIC DIAGRAM

PLS100/101

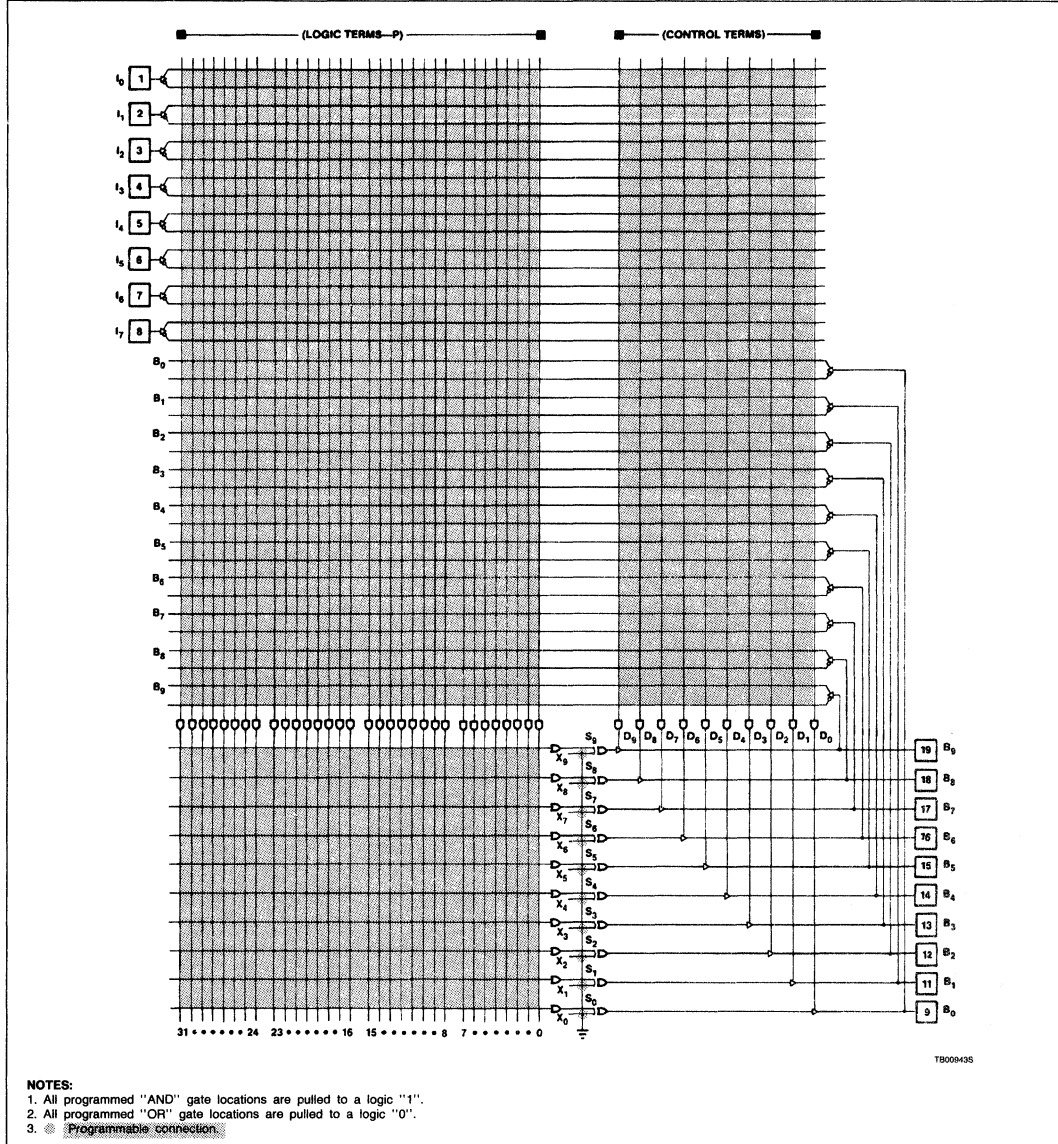


# Signetics Programmable Logic

# Introduction

## FPLA LOGIC DIAGRAM

PLS153/153A/PLHS153

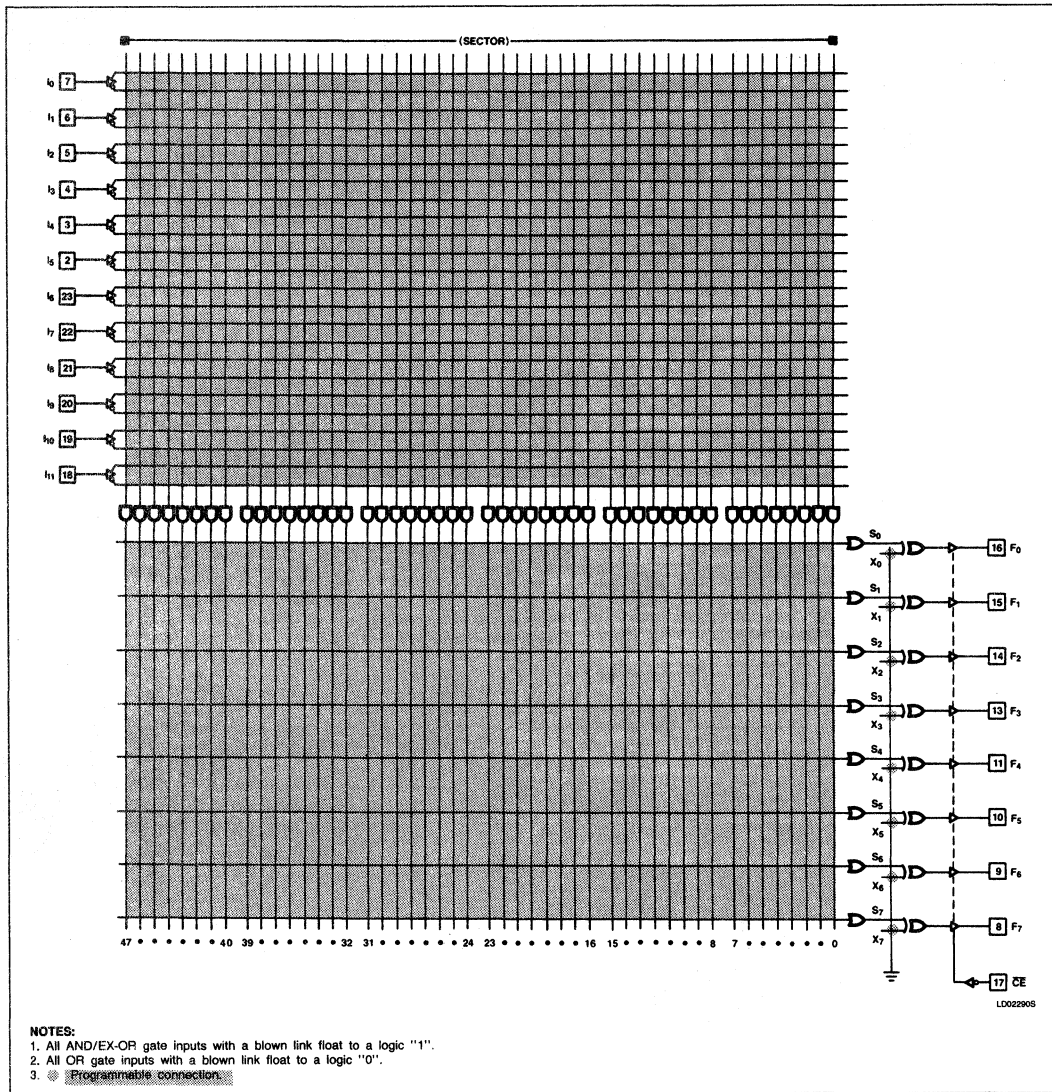


# Signetics Programmable Logic

# Introduction

## FPLA LOGIC DIAGRAM

PLS161

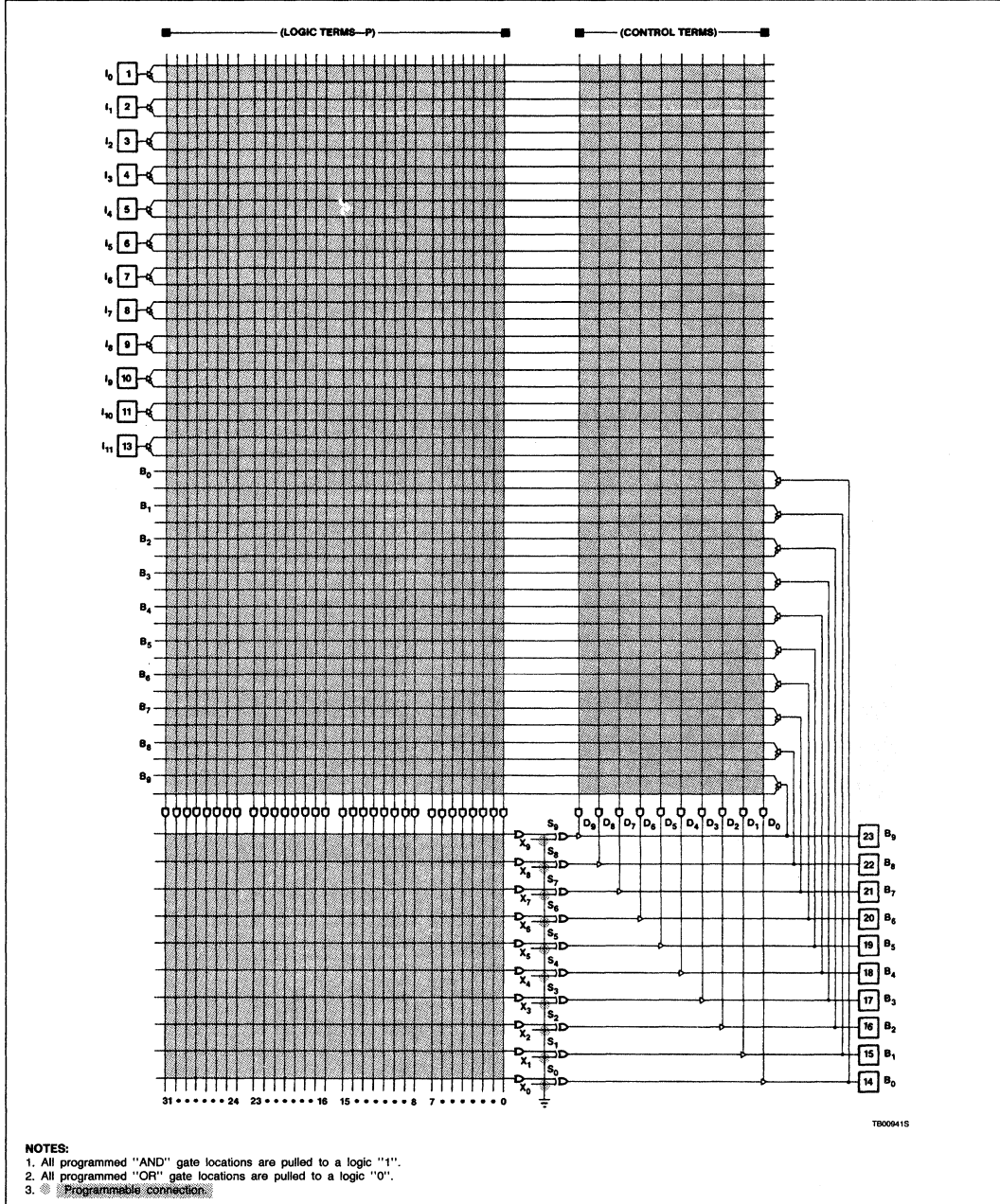


# Signetics Programmable Logic

# Introduction

## FPLA LOGIC DIAGRAM

PLS173/PLUS173

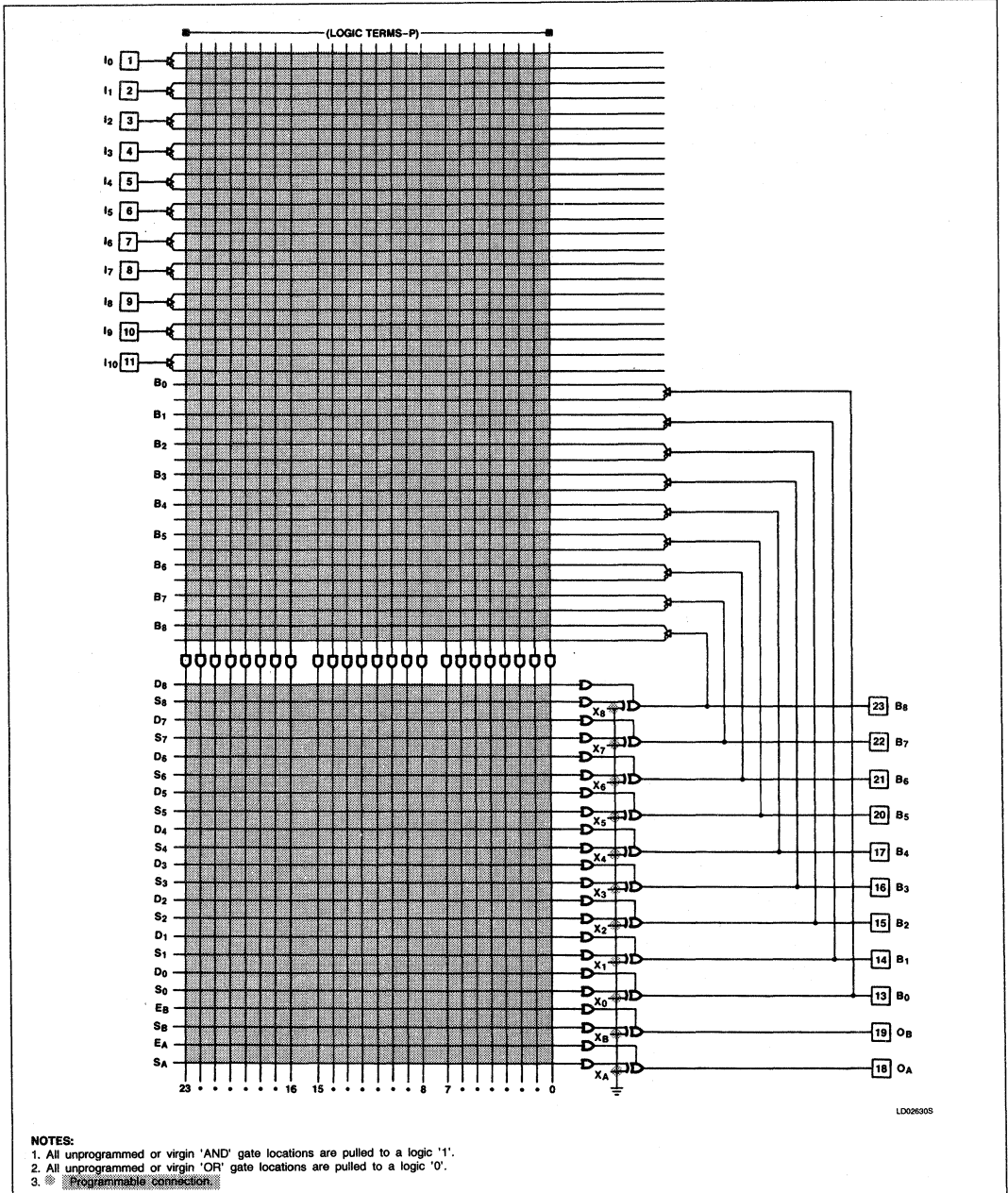


# Signetics Programmable Logic

# Introduction

## FPLA LOGIC DIAGRAM

PLHS473



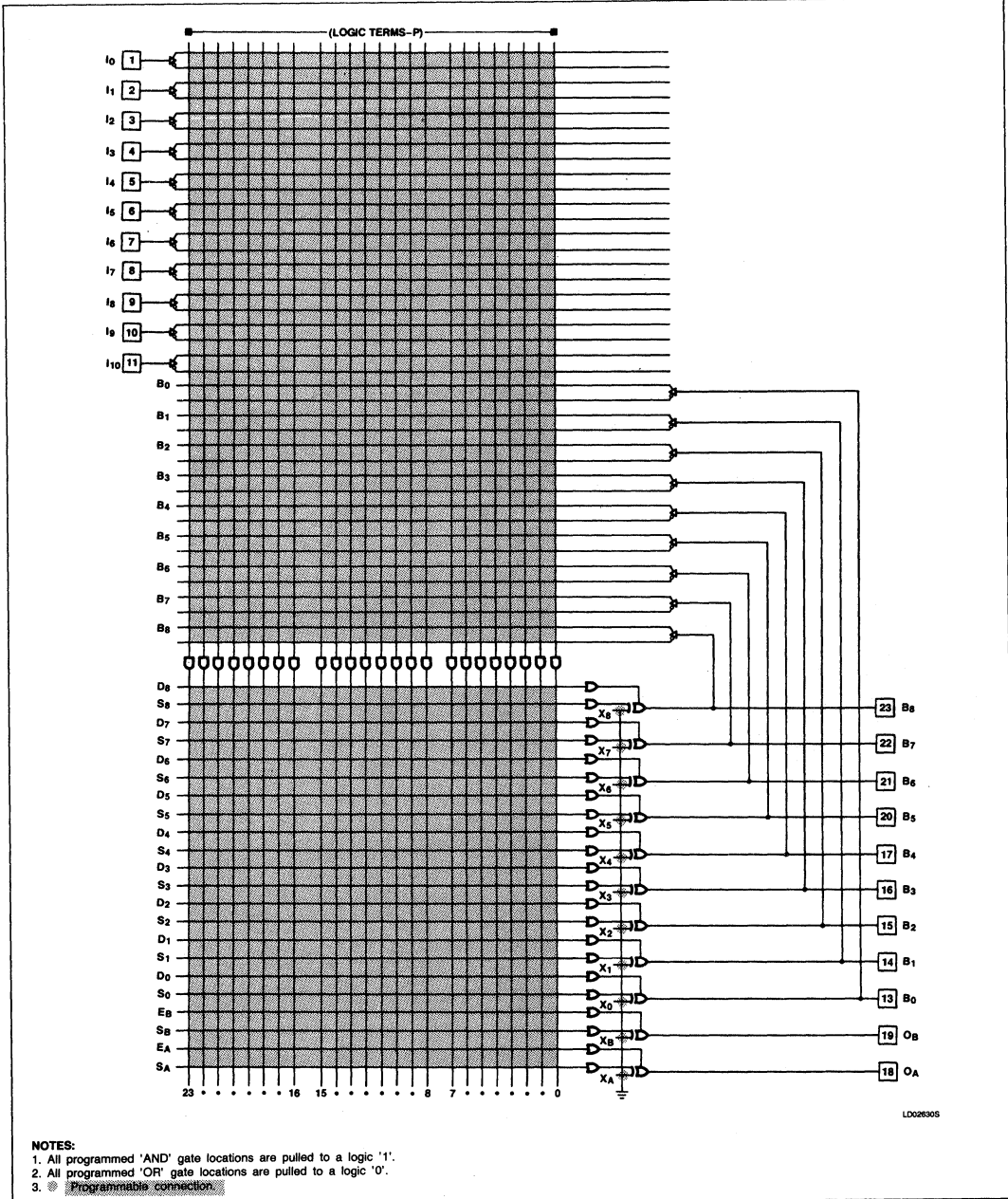
# Signetics Programmable Logic

# Introduction

## FPLA LOGIC DIAGRAM

PLC473

**2**



LD028305

**NOTES:**

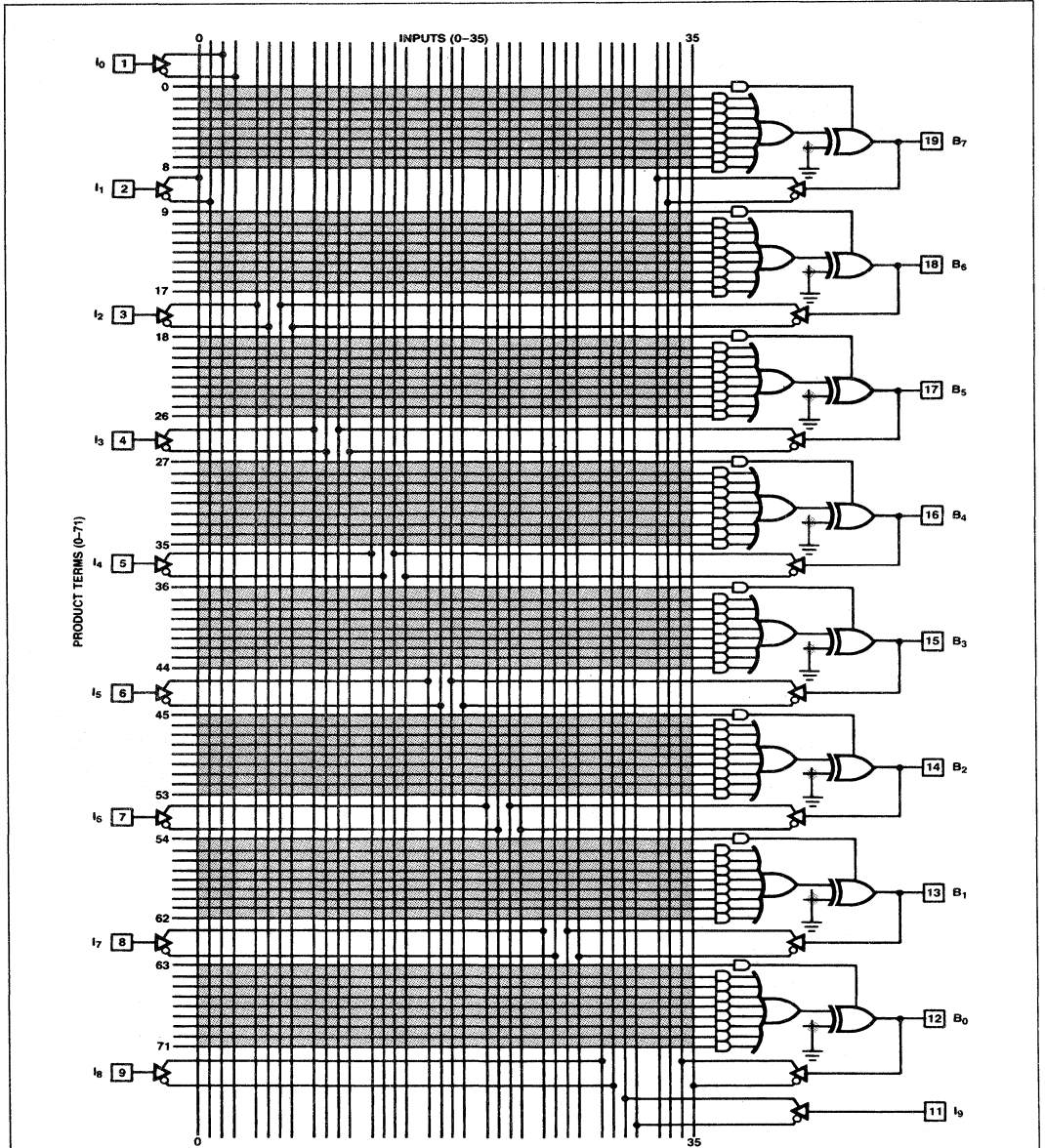
1. All programmed 'AND' gate locations are pulled to a logic '1'.
2. All programmed 'OR' gate locations are pulled to a logic '0'.
3. Programmable connection.

# Signetics Programmable Logic


# Introduction

## FPLA LOGIC DIAGRAM

PLHS18P8A



**NOTES:**

1. All unprogrammed or virgin "AND" gate locations are pulled to a logic "1".
2.  Programmable connections.

LD06205

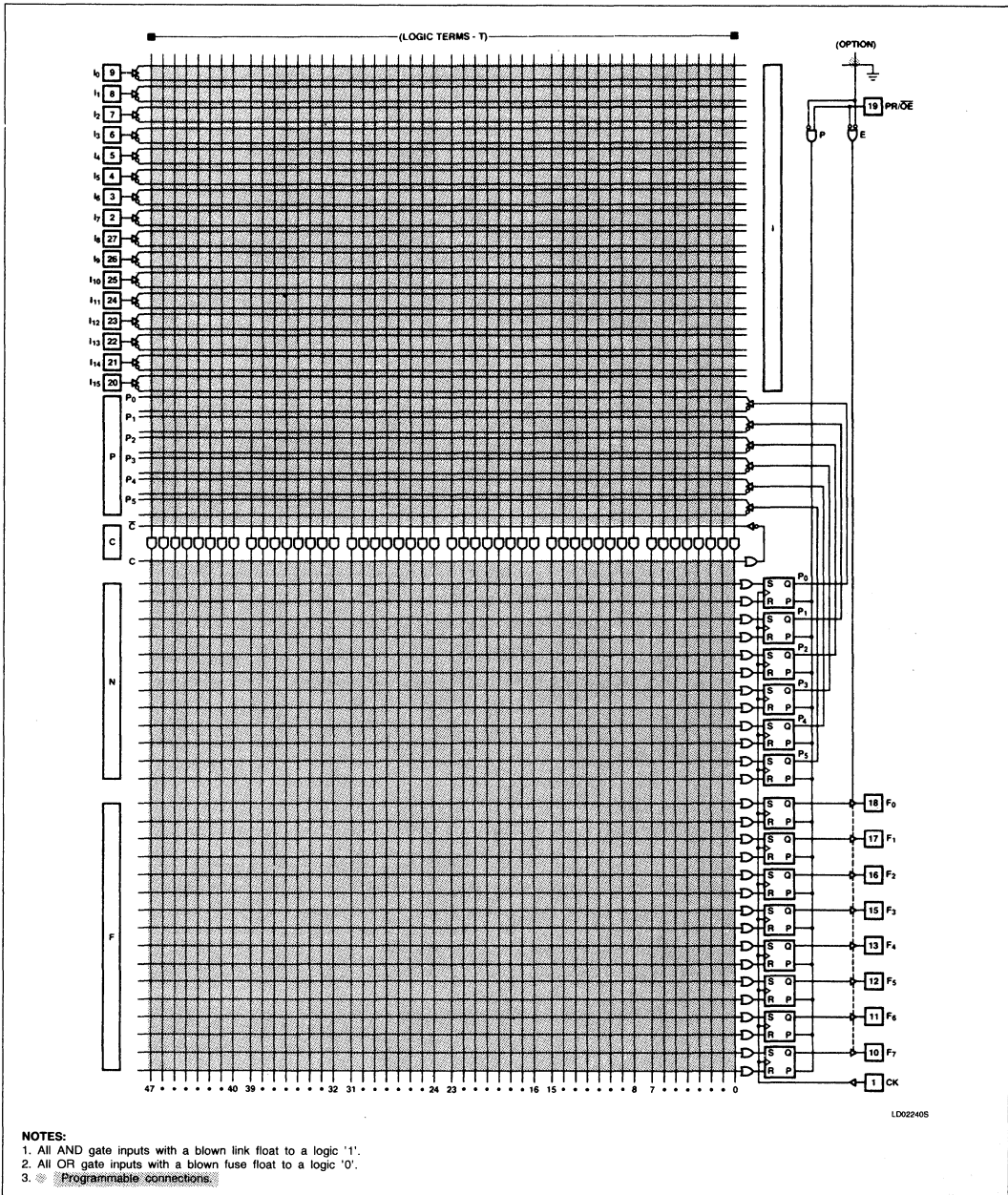


# Signetics Programmable Logic

# Introduction

## FPLS LOGIC DIAGRAM

PLS105/105A



**NOTES:**

1. All AND gate inputs with a blown link float to a logic '1'.
2. All OR gate inputs with a blown fuse float to a logic '0'.
3. Programmable connections.

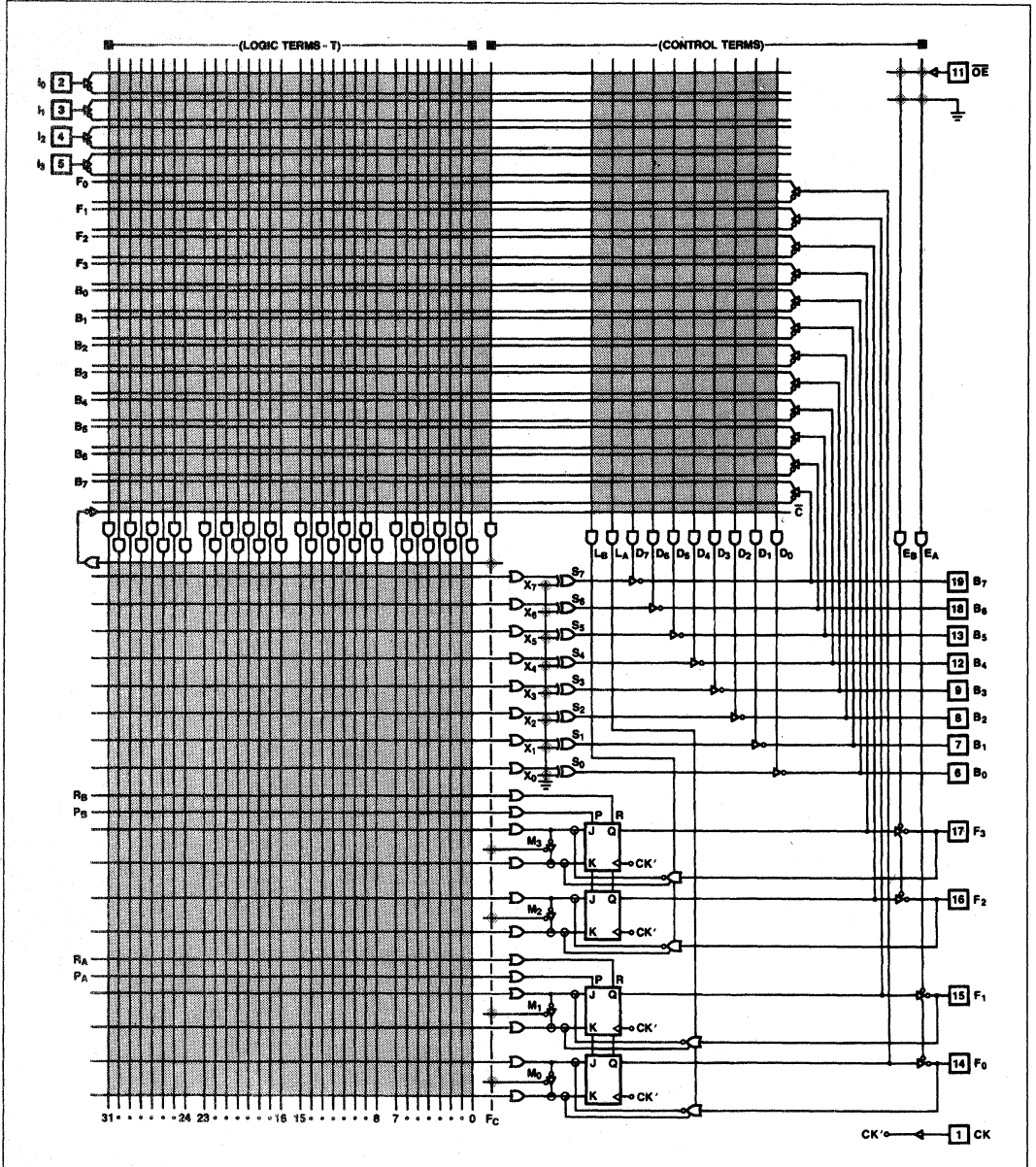


# Signetics Programmable Logic

# Introduction

## FPLS LOGIC DIAGRAM

PLS155



**NOTES:**

1. All OR gate inputs with a blown link float to a logic '0'.
2. All other gates and control inputs with a blown link float to a logic '1'.
3. ⊕ denotes WIRE-OR.
4. ⊙ Programmable connection.

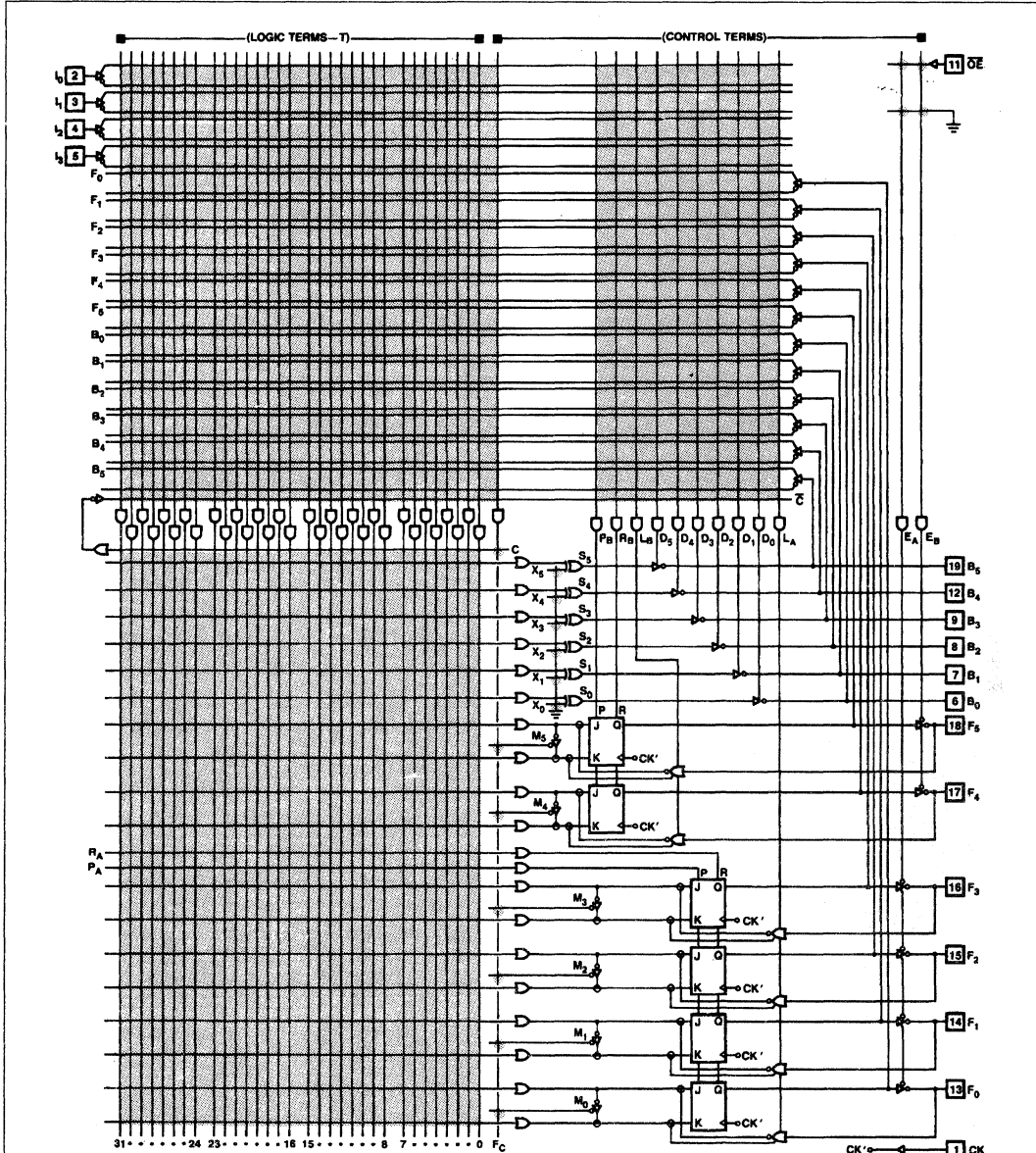
LD01702S

# Signetics Programmable Logic

# Introduction

## FPLS LOGIC DIAGRAM

PLS157



**NOTES:**

1. All OR gate inputs with a blown link float to a logic '0'.
2. All other gates and control inputs with a blown link float to a logic '1'.
3. ⊕ denotes WIRE-OR.
4. ⊙ denotes Programmable connections.

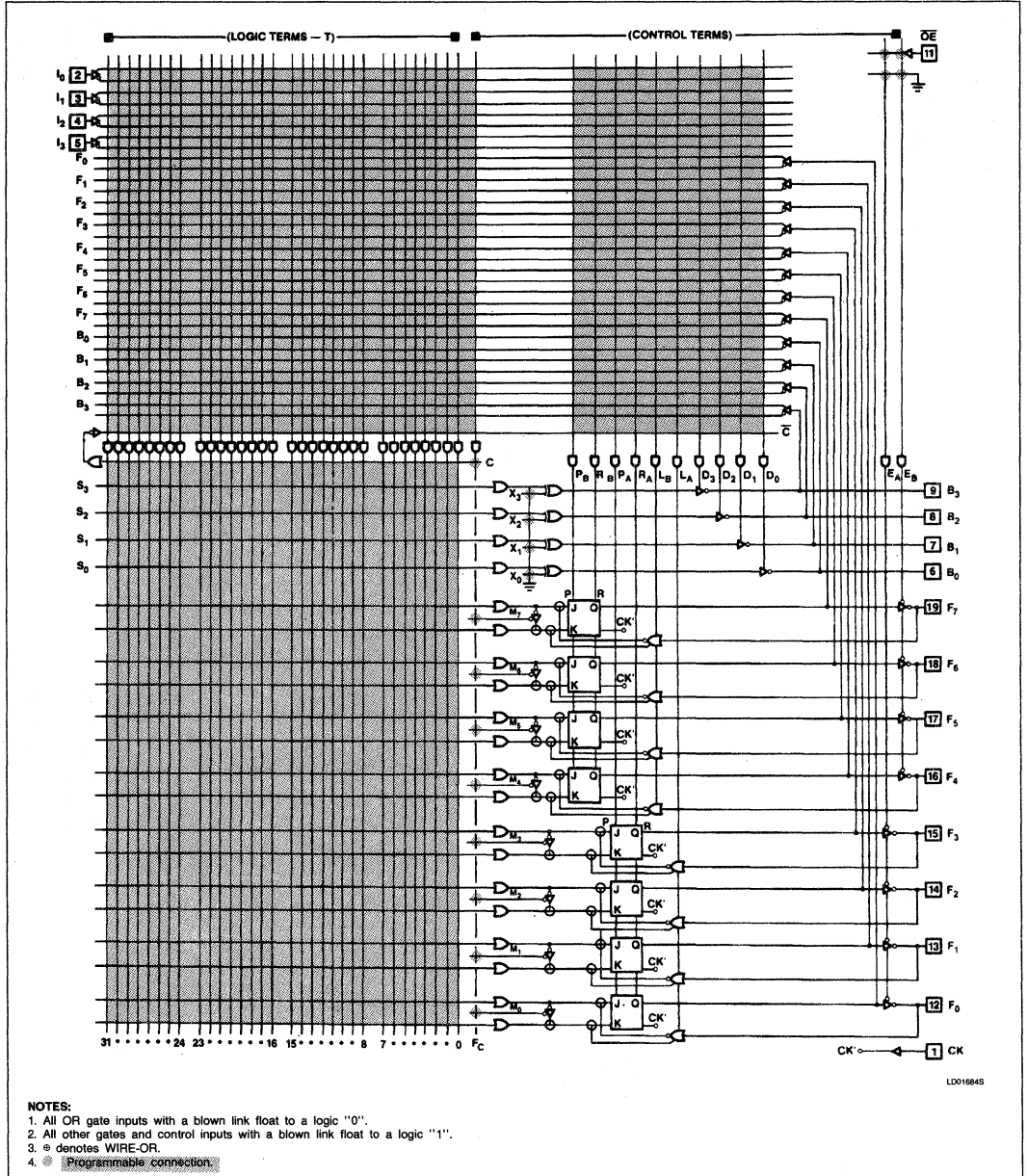
LDO1893S

# Signetics Programmable Logic

# Introduction

## FPLS LOGIC DIAGRAM

PLS159/159A

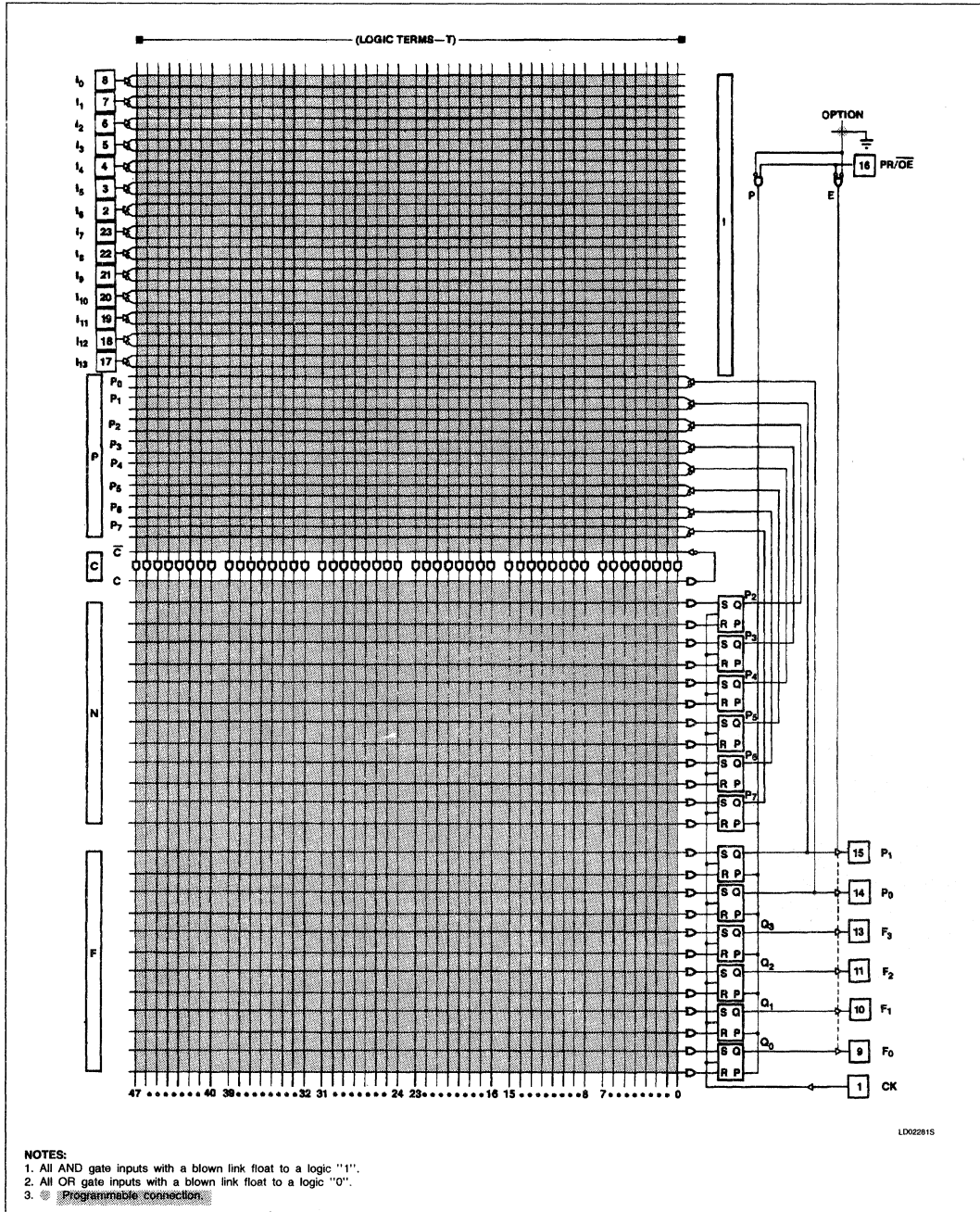


# Signetics Programmable Logic

# Introduction

## FPLS LOGIC DIAGRAM

PLS167/167A

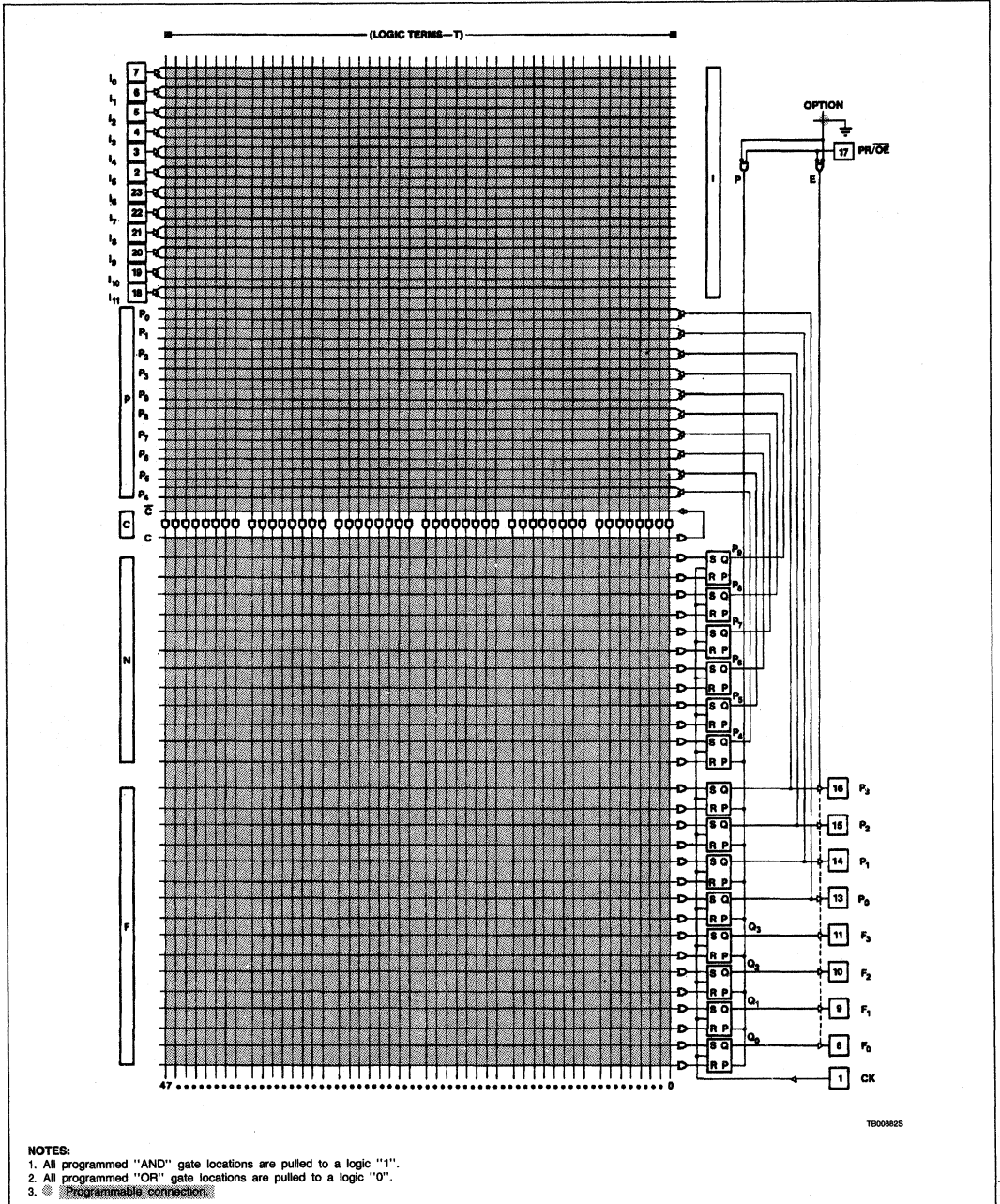


# Signetics Programmable Logic

# Introduction

## FPLS LOGIC DIAGRAM

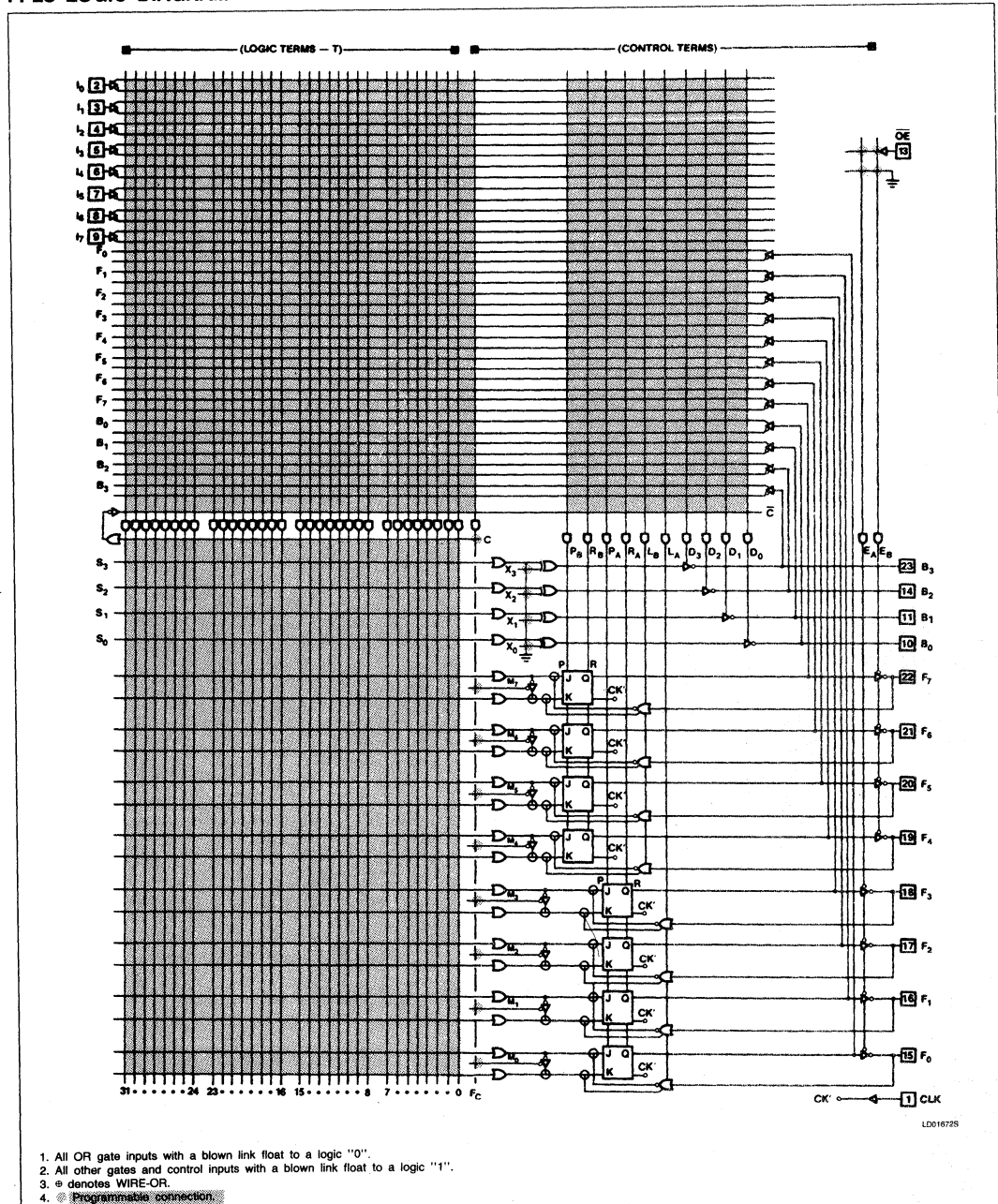
PLS168/168A



FPLS LOGIC DIAGRAM

PLS179

2



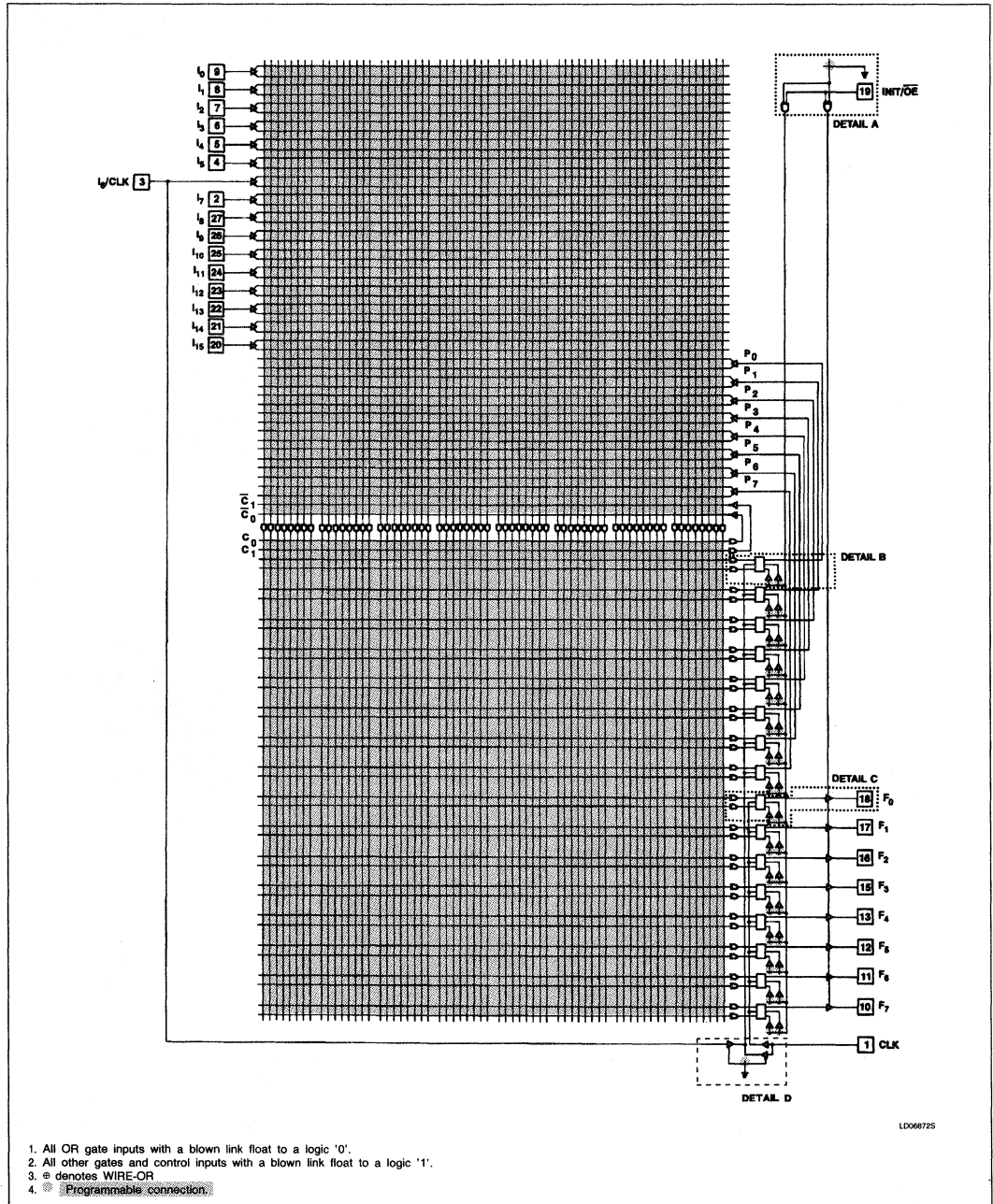


# Signetics Programmable Logic

# Introduction

## FPLS LOGIC DIAGRAM

PLUS405A

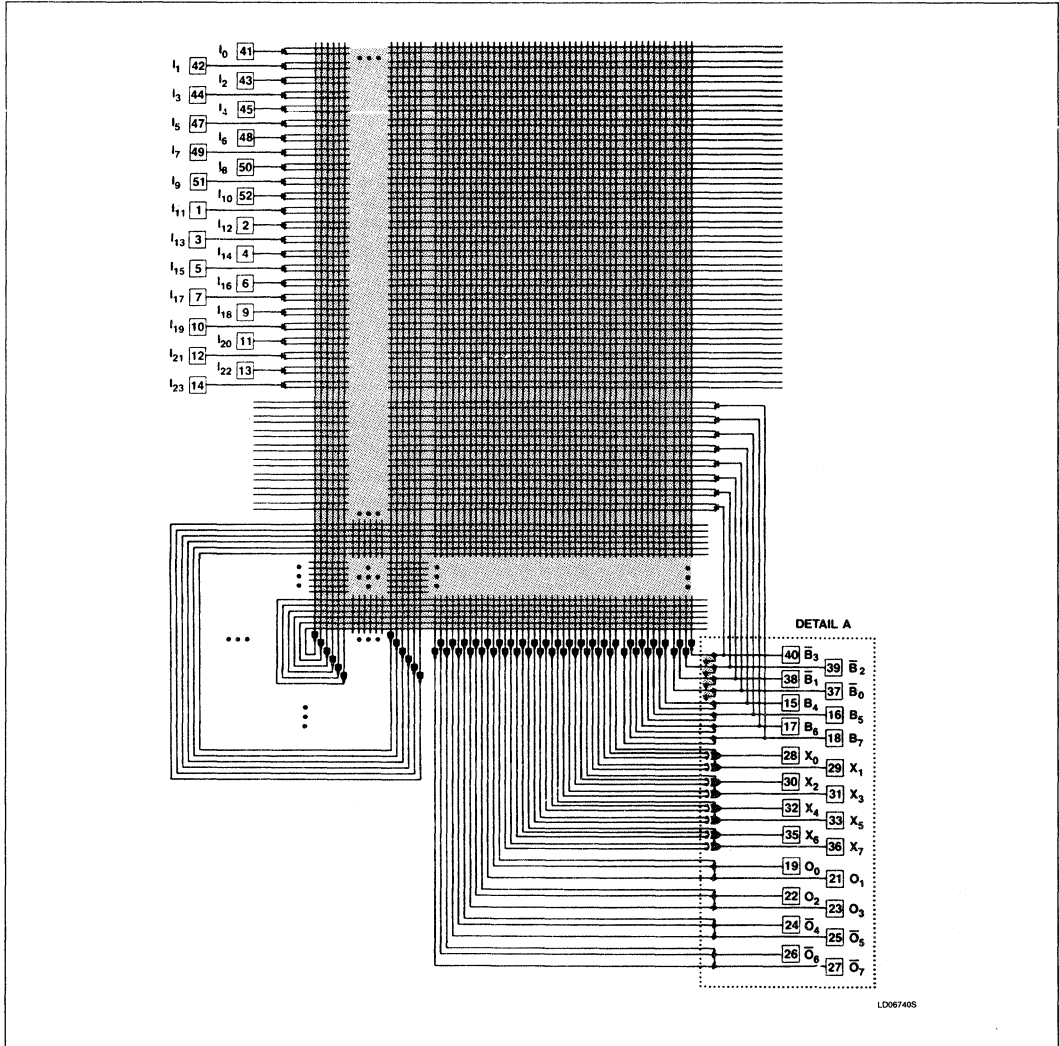


# Signetics Programmable Logic

# Introduction

## FPML LOGIC DIAGRAM

PLHS501



LD067405



## Signetics Programmable Logic

## Programming Information

---

### PLD PROGRAMMING

The current Signetics family of Programmable Logic (PLD) products are high-performance bipolar and CMOS devices which use various fusing elements to provide the many benefits of field programming. Programming is accomplished by application of voltages above those used for normal operation; therefore, no special pins are required for programming. The programming voltages and timing requirements make unintentional programming virtually impossible.

### PROGRAMMING INFORMATION

Complete programming system specifications for PLD 20-, 24-, and 28-pin and 52-pin PML products are available upon request from Application Specific Product Marketing.

Signetics encourages the purchase of programming equipment from a manufacturer who has a full-line of programming products to offer. Signetics also encourages the manufacturers of PLD programming equipment to submit their equipment for verification of electrical parameters and programming procedures. Information on manufacturers offering equipment certified by Signetics is available on request from Application Specific Product Marketing.

### SIGNETICS DISCOURAGES THE CONSTRUCTION AND USE OF "HOMEMADE" PROGRAMMING EQUIPMENT

In order to consistently achieve excellent programming yields, periodic calibration of the programming equipment is required. Consult the equipment manufacturer for the recommended calibration interval. Records of programming yield, by device type, should be kept and any downward trend or sudden change should be considered as an indication of a need to recalibrate the programming equipment.

### INDEX

#### Section 3 - Advanced Product Preview

PLMC153	Mask-Programmable Logic Array (18 × 42 × 10) .....	3-3
PLUS153	Field-Programmable Logic Array (18 × 42 × 10) .....	3-5
PLUS173	Field-Programmable Logic Array (22 × 42 × 10) .....	3-7
PLHS501	Programmable Macro Logic Random Logic Unit (32 × 72 × 24) ...	3-9
PLHS502	Programmable Macro Logic Synchronous/Asynchronous Register Macro Unit (32 × 64 × 24) .....	3-19



## PLMC153 Mask-Programmable Logic Array (18 × 42 × 10)

Signetics Programmable Logic  
*Objective Specification*

### Application Specific Products • Series 20

#### DESCRIPTION

The PLMC153 is a CMOS, mask-programmable functional equivalent of the PLS153 and PLS153A FPLAs. Custom logic patterns can be generated directly from working bipolar PLS153/153A devices. This enables the user to prototype and debug a device in a system prior to committing the pattern to a mask set. The programmed PLMC153 provides maximum production economy, while the CMOS process technology reduces the power consumption of the PLMC153 to less than one third of its bipolar counterparts.

Identical to the PLS153 and 153A, the PLMC153 is a two-level logic element, consisting of 42 AND gates and 10 OR gates, featuring programmable I/O polarity and direction.

All AND gates are linked to 8 inputs (I) and 10 bidirectional I/O lines (B). These yield variable I/O gate configurations via 10 direction control gates (D), ranging from 18 inputs to 10 outputs.

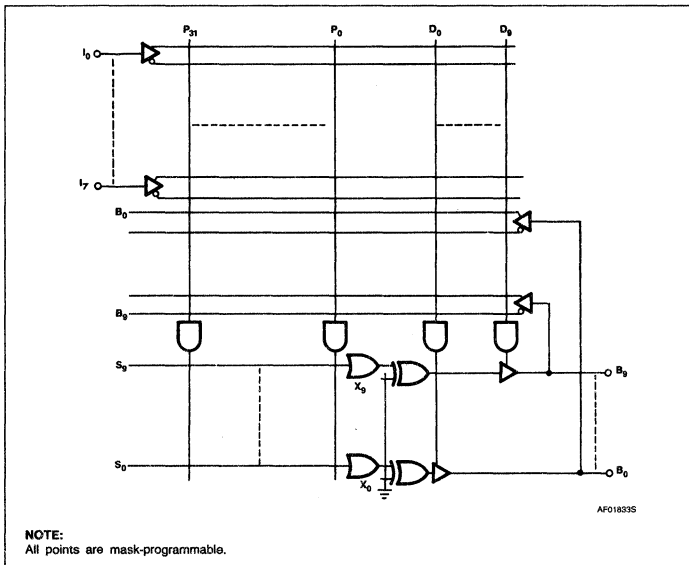
On chip T/C buffers couple either True (I, B) or Complement ( $\bar{I}$ ,  $\bar{B}$ ) input polarities to all AND gates, whose outputs can

be optionally linked to all OR gates. Their output polarity, in turn, is individually programmable through a set of EX-OR gates for implementing AND/OR or AND/NOR logic functions.

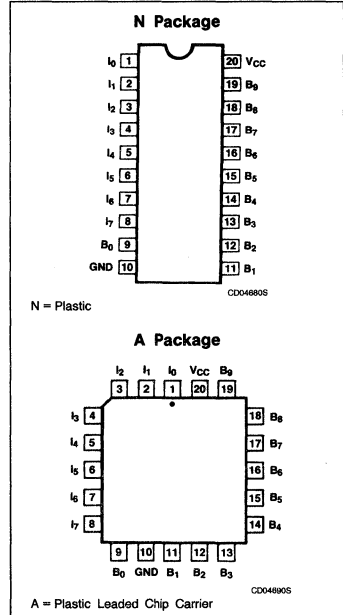
#### FEATURES

- Single Mask-Programmable
- Functionally identical to PLS153 and PLS153A
- 8 inputs
- 42 AND gates
- 10 OR gates
- 10 bidirectional I/O lines
- Active-High or -Low outputs
- 42 product terms:
  - 32 logic terms
  - 10 control terms
- I/O propagation delay: 35ns (max.)
- Power dissipation: (typ.)
  - DC: 1mA
  - AC: 1mA/MHz
- Input Loading:  $-100\mu\text{A}$  (max.)
- Tri-state outputs
- TTL compatible

#### FUNCTIONAL DIAGRAM



#### PIN CONFIGURATIONS



#### APPLICATIONS

- Random logic
- Code converters
- Fault detectors
- Function generators
- Address mapping
- Multiplexing

#### LOGIC FUNCTION

TYPICAL PRODUCT TERM:

$$P_n = A \cdot B \cdot C \cdot D \cdot \dots$$

TYPICAL LOGIC FUNCTION:

AT OUTPUT POLARITY = H

$$Z = P_0 + P_1 + P_2 \dots$$

AT OUTPUT POLARITY = L

$$Z = \bar{P}_0 + \bar{P}_1 + \bar{P}_2 \dots$$

$$Z = \bar{P}_0 \cdot \bar{P}_1 \cdot \bar{P}_2 \dots$$

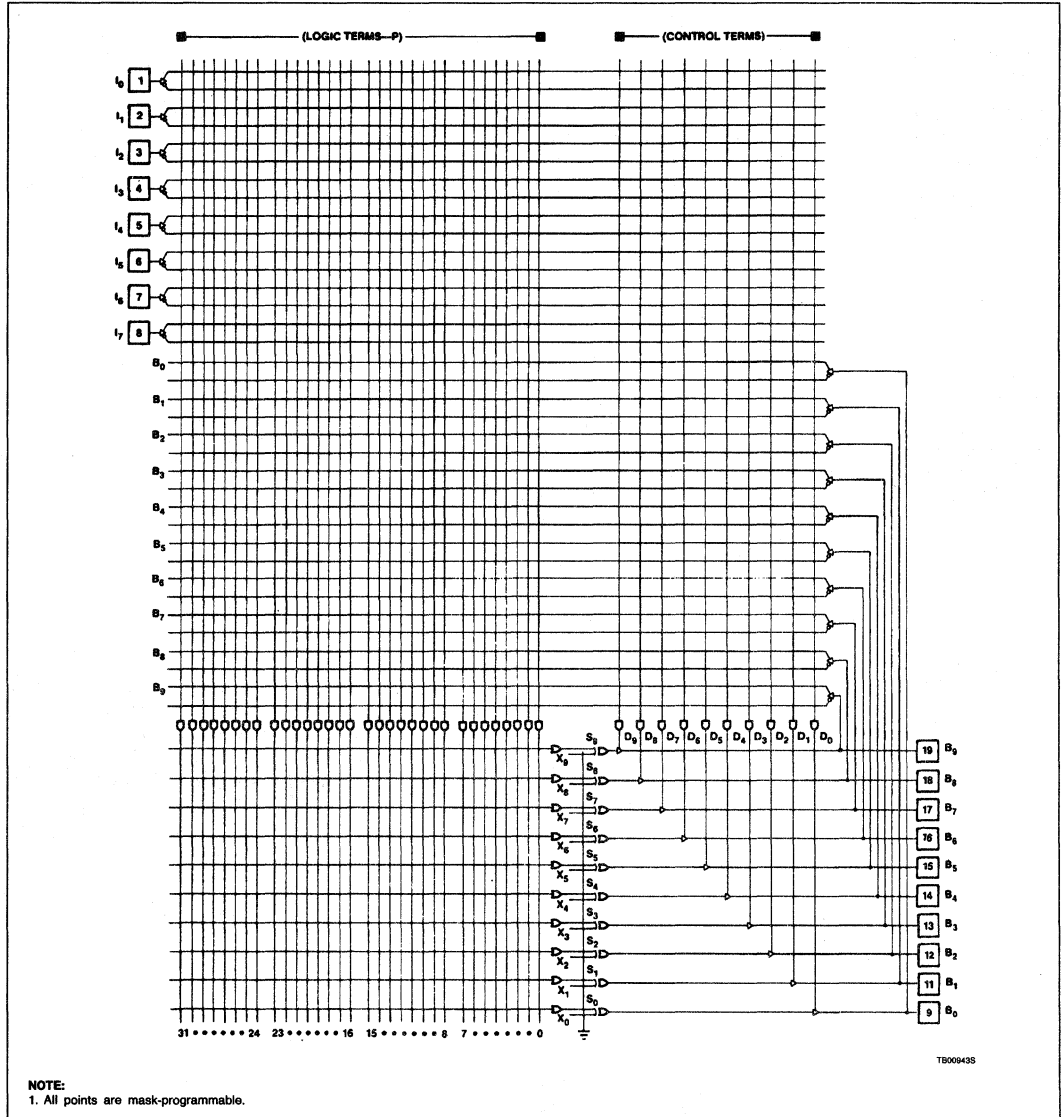
#### NOTES:

1. For each of the 10 outputs, either function Z (Active-High) or  $\bar{Z}$  (Active-Low) is available, but not both. The desired output polarity is programmed via the EX-OR gates.
2. Z, A, B, C, etc. are user defined connections to fixed inputs (I) and bidirectional pins (B).

# Mask-Programmable Logic Array (18 × 42 × 10)

PLMC153

## FPLS LOGIC DIAGRAM



## PLUS153 Field-Programmable Logic Array (18 × 42 × 10)

Signetics Programmable Logic  
*Objective Specification*

### Application Specific Products • Series 20

#### DESCRIPTION

The PLUS153 is a two-level logic element, consisting of 42 AND gates and 10 OR gates with fusible link connections for programming I/O polarity and direction.

All AND gates are linked to 8 inputs (I) and 10 bidirectional I/O lines (B). These yield variable I/O gate configurations via 10 direction control gates (D), ranging from 18 inputs to 10 outputs.

On chip T/C buffers couple either True (I, B) or Complement ( $\bar{I}$ ,  $\bar{B}$ ) input polarities to all AND gates, whose outputs can be optionally linked to all OR gates. Their output polarity, in turn, is individually programmable through a set of EX-OR gates for implementing AND/OR or AND/NOR logic functions.

The PLUS153 is field programmable, enabling the user to quickly generate custom patterns using standard programming equipment.

Order codes are contained on the pages following.

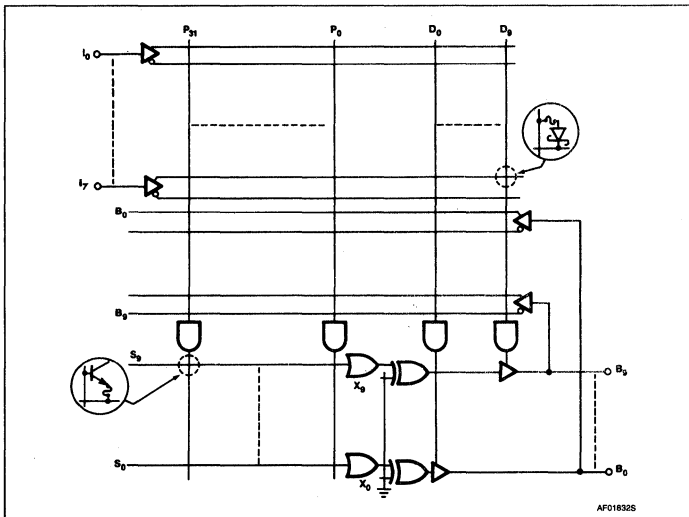
#### FEATURES

- Field-Programmable (TI-W links)
- Functionally identical to PLS153/153A and PLHS153
- 8 inputs
- 42 AND gates
- 10 OR gates
- 10 bidirectional I/O lines
- Active-High or -Low outputs
- 42 product terms:
  - 32 logic terms
  - 10 control terms
- I/O propagation delay: 12ns (max.)
- Input loading: -100 $\mu$ A (max.)
- Power dissipation: 700mW (typ.)
- Tri-state outputs
- TTL compatible

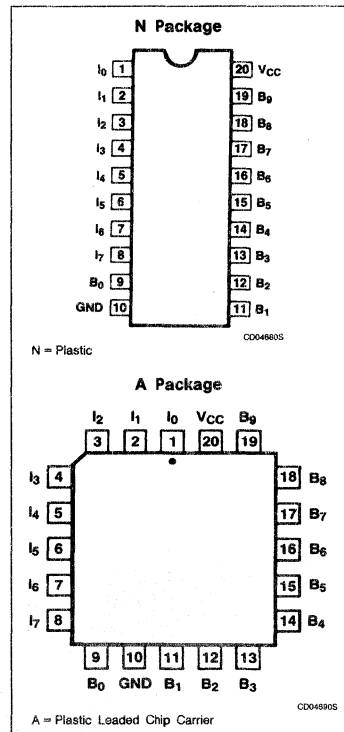
#### APPLICATIONS

- Random logic
- Code converters
- Fault detectors
- Function generators
- Address mapping
- Multiplexing

#### FUNCTIONAL DIAGRAM



#### PIN CONFIGURATIONS



#### LOGIC FUNCTION

TYPICAL PRODUCT TERM:  
 $P_n = A \cdot B \cdot C \cdot D \cdot \dots$   
TYPICAL LOGIC FUNCTION:  
AT OUTPUT POLARITY = H  
 $Z = P_0 + P_1 + P_2 \dots$   
AT OUTPUT POLARITY = L  
 $Z = P_0 \cdot P_1 \cdot P_2 \dots$

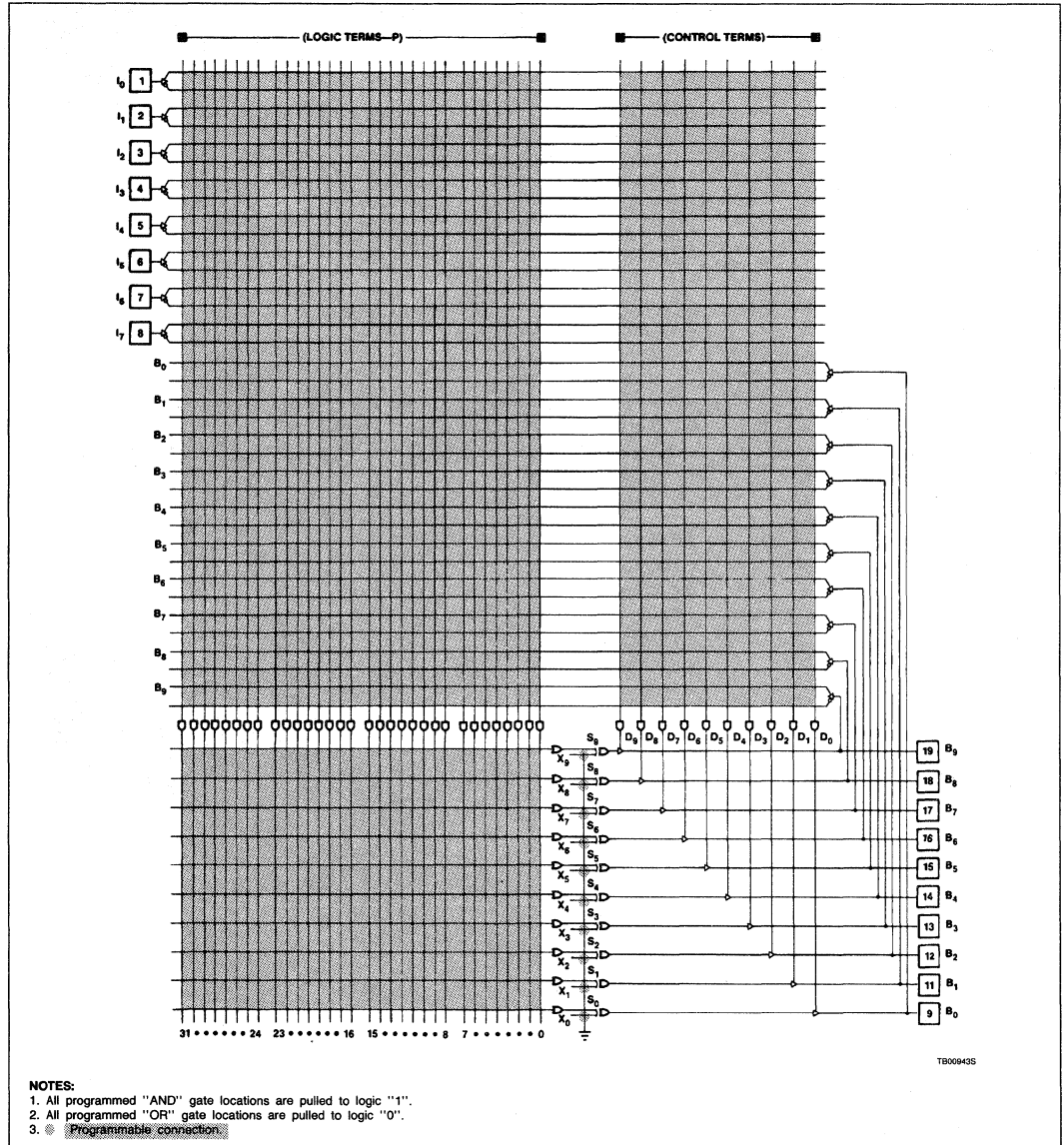
#### NOTES:

1. For each of the 10 outputs, either function Z (Active-High) or  $\bar{Z}$  (Active-Low) is available, but not both. The desired output polarity is programmed via the EX-OR gates.
2. Z, A, B, C, etc. are user defined connections to fixed inputs (I) and bidirectional pins (B).

# Field-Programmable Logic Array (18 × 42 × 10)

## PLUS153

### FPLA LOGIC DIAGRAM



## PLUS173 Field-Programmable Logic Array (22 × 42 × 10)

Signetics Programmable Logic  
*Objective Specification*

### Application Specific Products • Series 24

#### DESCRIPTION

The PLUS173 is a high-speed version of the PLS173 FPLA. The Signetics state-of-the-art Oxide Isolated Bipolar fabrication process is employed to produce performance levels not yet achieved in devices of this complexity.

The PLUS173 is a two-level logic element consisting of 42 AND gates and 10 OR gates with fusible link connections for programming I/O polarity and direction.

All AND gates are linked to 12 inputs (I) and 10 bidirectional I/O lines (B). These yield variable I/O gate configurations via 10 direction control gates (D), ranging from 22 inputs to 10 outputs.

On chip T/C buffers couple either True (I, B) or Complement (I,  $\bar{B}$ ) input polarities to all AND gates, whose outputs can be optionally linked to all OR gates. Their output polarity, in turn, is individually programmable through a set of EX-OR gates for implementing AND/OR or AND/NOR logic functions.

The PLUS173 is field-programmable, enabling the user to quickly generate custom patterns using standard programming equipment.

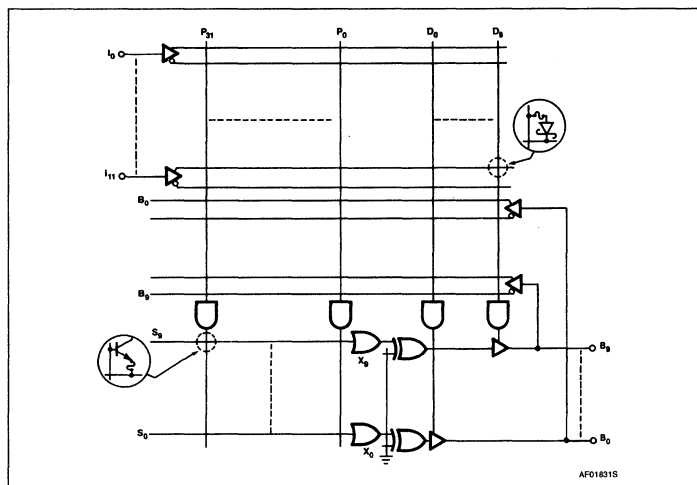
#### FEATURES

- Field-Programmable (Ti-W links)
- Functionally identical to PLS173
- 12 inputs
- 42 AND gates
- 10 OR gates
- 10 bidirectional I/O lines
- Active-High or -Low outputs
- 42 product terms:
  - 32 logic terms
  - 10 control terms
- I/O propagation delay: 12ns (max.)
- Input loading: - 100 $\mu$ A (max.)
- Power dissipation: 700mW (typ.)
- Tri-state outputs
- TTL compatible

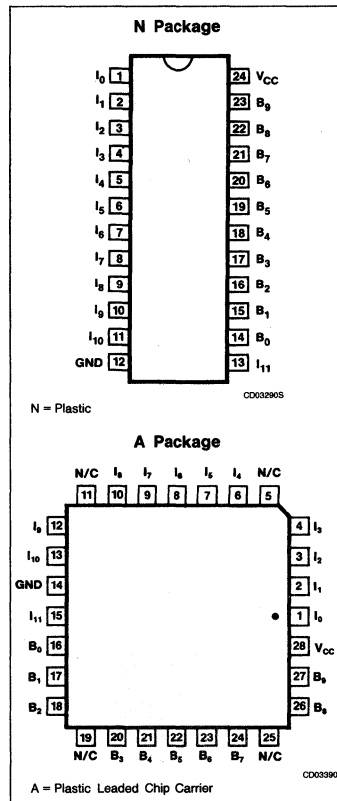
#### APPLICATIONS

- Random logic
- Code converters
- Fault detectors
- Function generators
- Address mapping
- Multiplexing

#### FUNCTIONAL DIAGRAM



#### PIN CONFIGURATIONS



#### LOGIC FUNCTION

**TYPICAL PRODUCT TERM:**

$$P_n = A \cdot B \cdot C \cdot D \dots$$

**TYPICAL LOGIC FUNCTION:**

**AT OUTPUT POLARITY = H**

$$Z = P_0 + P_1 + P_2 \dots$$

**AT OUTPUT POLARITY = L**

$$Z = P_0 + P_1 + P_2 + \dots$$

$$Z = P_0 \cdot P_1 \cdot P_2 \dots$$

#### NOTES:

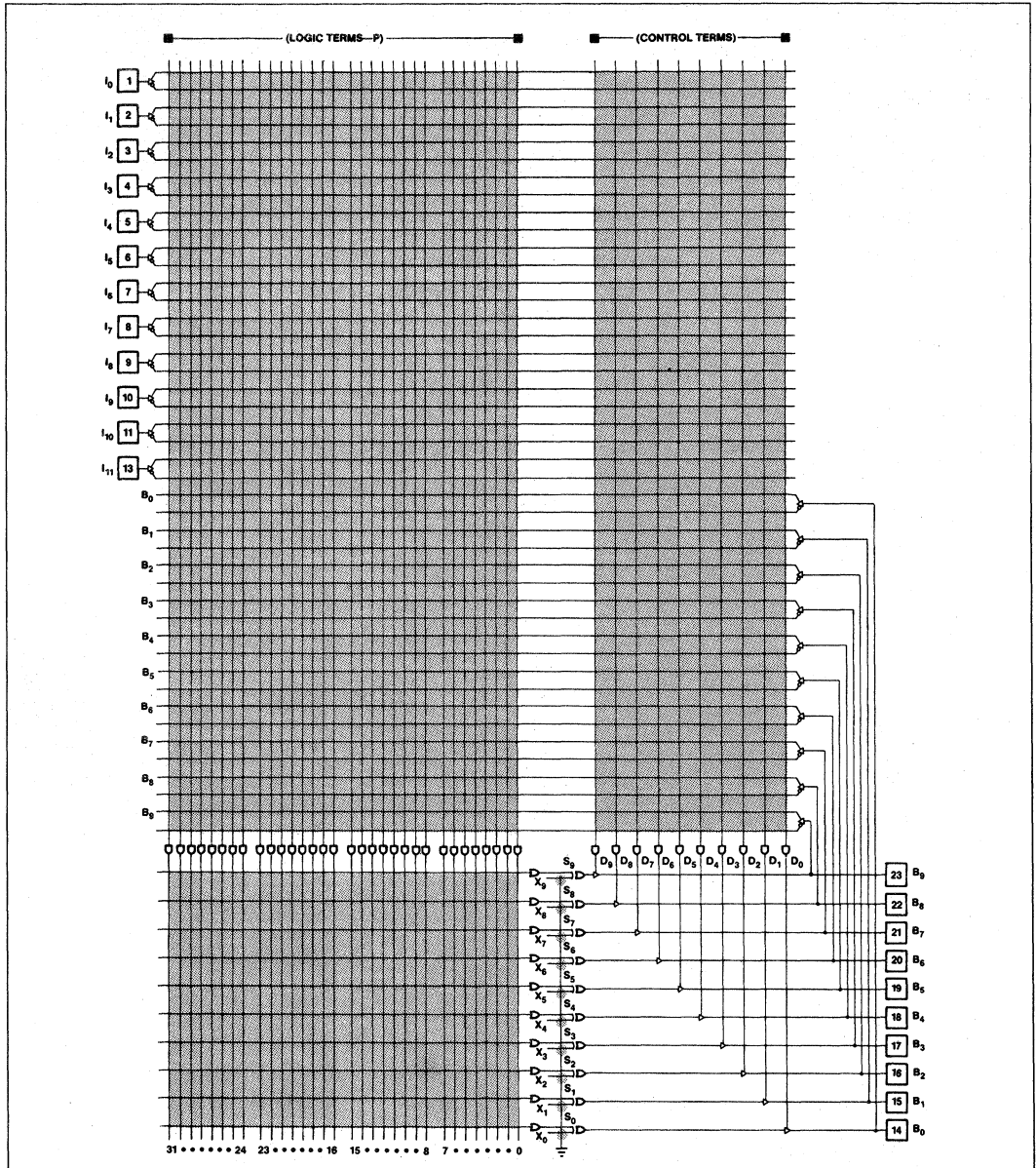
1. For each of the 10 outputs, either function Z (Active-High) or  $\bar{Z}$  (Active-Low) is available, but not both. The desired output polarity is programmed via the EX-OR gates.
2. Z, A, B, C, etc. are user defined connections to fixed inputs (I) and bidirectional pins (B).



# Field-Programmable Logic Array (22 × 42 × 10)

PLUS173

## FPLA LOGIC DIAGRAM



**NOTES:**

1. All programmed 'AND' gate locations are pulled to logic "1".
2. All programmed 'OR' gate locations are pulled to logic "0".
3. ● Programmable connection.

TB009415

# PLHS501

## Programmable Macro Logic Random Logic Unit\*

### (32 × 72 × 24)

**Application Specific Products**  
● PML Series 52

*Preliminary Specification*

### DESCRIPTION

The PLHS501 is a member of the Signetics Programmable Macro Logic family. PML is unique in its capability of performing other than two level logic functions without incurring I/O buffer delays. This allows the logic or system designer to imbed logical operations or macro structures within the framework of the I/O pins. Since the imbedded functions are independent of the delays created by the I/O buffers, they can be performed at speeds lesser architectures cannot reproduce.

The technique used to perform this operation is a NAND foldback network which allows the direct interconnection of any number of logic nodes within the single fuse matrix. Macros can be formed and then interconnected to the I/O structure. In addition, single-level and multi-level logic can be performed at speeds which reflect only the logic path utilized. Therefore, a single-level logic function has a very short path through the device. Additional levels incur only one NAND foldback delay per level. This delay is less than the combined delay created by previous generations of devices which stipulate that the logic signal must pass through I/O buffers after one or two levels of logic are performed.

The PLHS501 is fabricated with Signetics, ZA Oxide-Isolated Bipolar Process. ZA utilizes Vertical Avalanche Migration Programmed (VAMP™) Fuses as programming elements. These fuses provide high programming yield and reliability. Proprietary onboard test circuitry allows the PLHS501 to be thoroughly tested prior to programming.

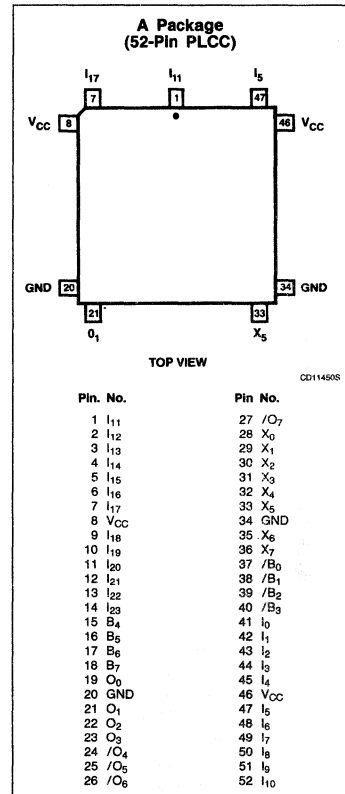
### FEATURES

- Signetics NAND foldback architecture
- Field-Programmable
- 24 dedicated inputs
- Fixed and programmable output buffers
  - 8 I/O buffers
  - 8 EX-OR buffers
  - 4 Active-Low buffers
  - 4 Active-High buffers
- 72 internal NAND foldback terms
- Supported by AMAZE Development System
- Testable in unprogrammed state
- Verify Lock Fuse
- TTL compatible
- Power dissipation: 1.25W (typ.)
- Logic delay times
  - Single-level = 18ns (max.)
  - Two-level = 26ns (max.)
  - Internal NAND delay = 8ns (max.)

### ARCHITECTURE

- 24 dedicated inputs;  $I_0 - I_{23}$
- 4 Active-High I/Os with individual enable:  $B_4 - B_7$
- 4 Active-Low I/Os with individual fused enable:  $\bar{B}_0 - \bar{B}_3$
- 2 Active-High output pairs; each pair with common enable:  $O_0 - O_3$
- 2 Active-Low output pairs; each pair with common enable:  $\bar{O}_4 - \bar{O}_7$
- 4 EX-OR output pairs; each pair with common enable:  $X_0 - X_7$
- 72 internal NAND foldback terms

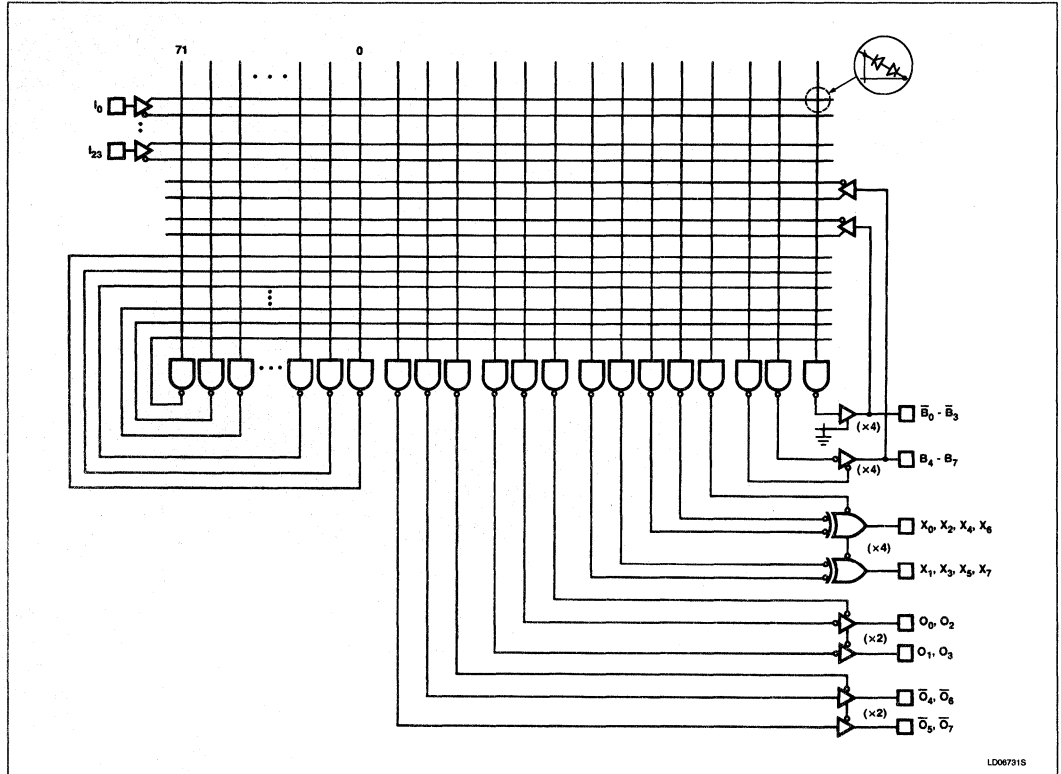
### PIN CONFIGURATION



# Programmable Macro Logic Random Logic Unit\* (32 X 72 X 24)

PLHS501

## FUNCTIONAL DIAGRAM



### ORDERING INFORMATION

DESCRIPTION	ORDER CODE
52-Pin Plastic Leaded Chip Carrier	PLHS501A

### THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

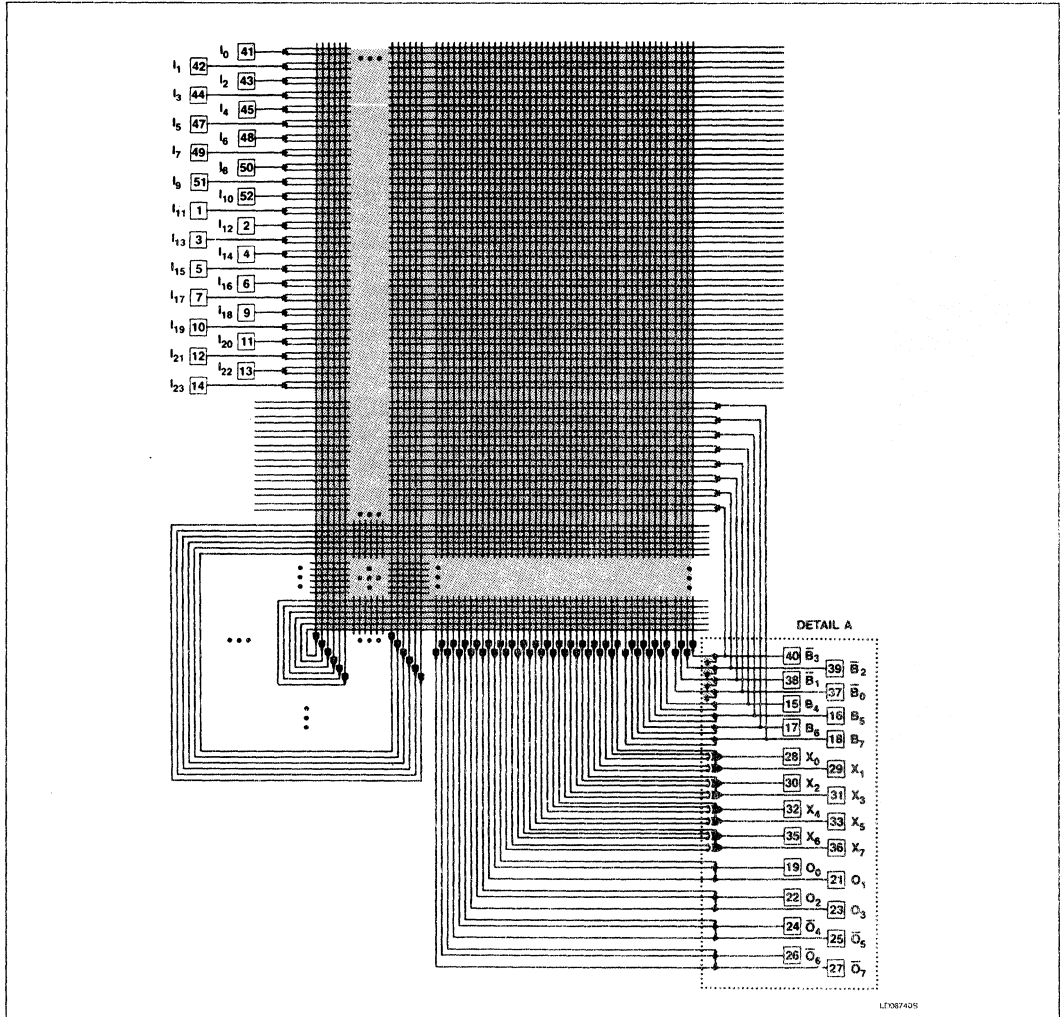
### ABSOLUTE MAXIMUM RATINGS<sup>1</sup>

SYMBOL	PARAMETER	RATING		UNIT
		Min	Max	
V <sub>CC</sub>	Supply voltage		+7	V <sub>DC</sub>
V <sub>IN</sub>	Input voltage		+5.5	V <sub>DC</sub>
V <sub>OUT</sub>	Output voltage		+5.5	V <sub>DC</sub>
I <sub>IN</sub>	Input currents	-30	+30	mA
I <sub>OUT</sub>	Output currents		+100	mA
T <sub>A</sub>	Temperature range			
T <sub>STG</sub>	Operating	0	+75	°C
	Storage	-65	+150	

# Programmable Macro Logic Random Logic Unit\* (32 X 72 X 24)

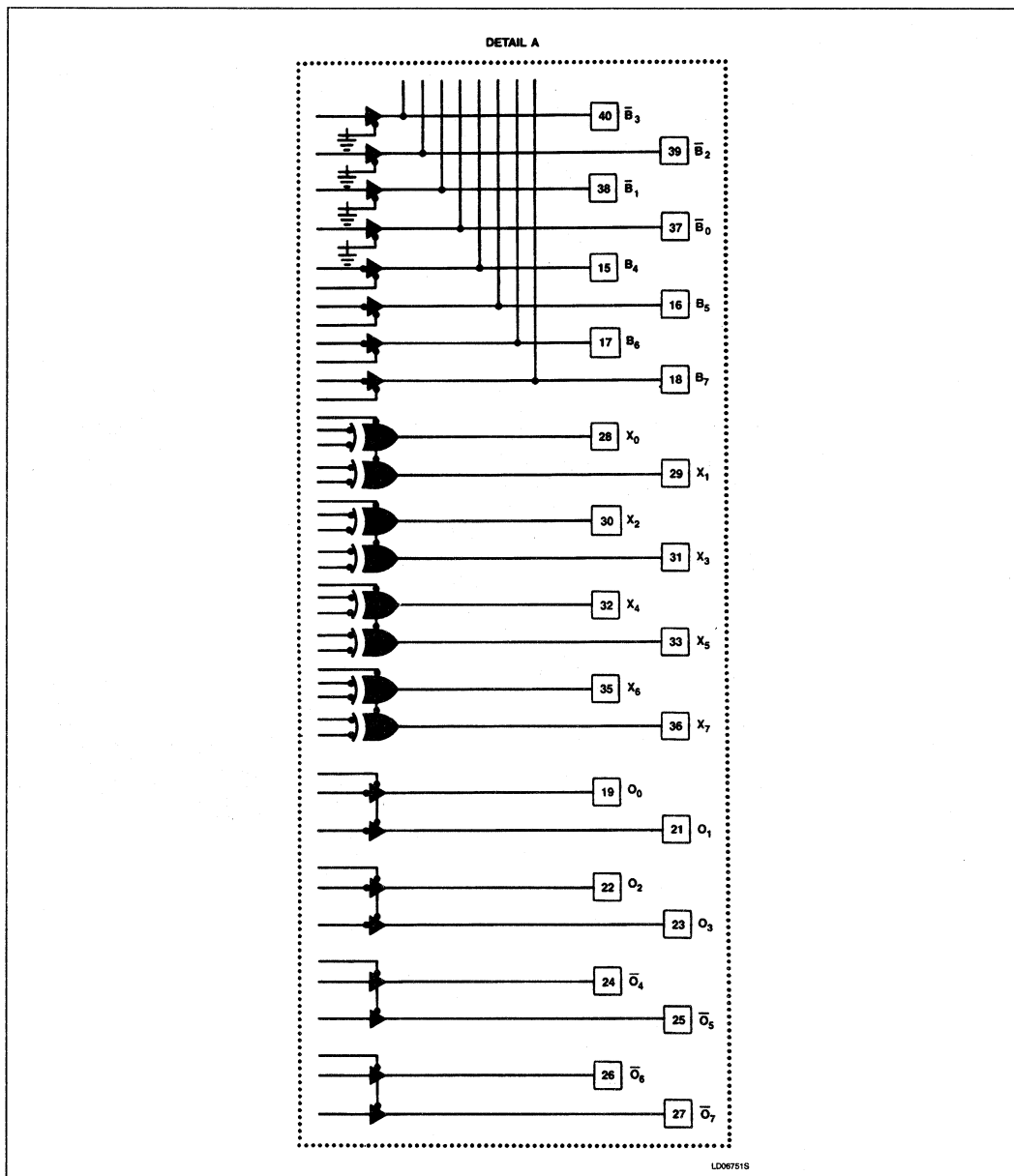
PLHS501

## LOGIC DIAGRAM



# Programmable Macro Logic Random Logic Unit\* (32 × 72 × 24)

PLHS501



# Programmable Macro Logic Random Logic Unit\* (32 × 72 × 24)

PLHS501

## DC ELECTRICAL CHARACTERISTICS 0°C ≤ T<sub>A</sub> ≤ +75°C, 4.75 ≤ V<sub>CC</sub> ≤ 5.25V

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ <sup>2</sup>	Max	
<b>Input voltage<sup>3</sup></b>						
V <sub>IL</sub> V <sub>IH</sub> V <sub>IC</sub>	Low High Clamp <sup>3,4</sup>	V <sub>CC</sub> = Min V <sub>CC</sub> = Max V <sub>CC</sub> = Min, I <sub>IN</sub> = -12mA	2.0	-0.8	0.8 -1.2	V
<b>Output voltage</b>						
V <sub>OL</sub> V <sub>OH</sub>	Low <sup>3,5</sup> High <sup>3,6</sup>	V <sub>CC</sub> = Min I <sub>OL</sub> = 10mA I <sub>OH</sub> = -2mA	2.4		0.45	V
<b>Input current</b>						
I <sub>IL</sub> I <sub>IH</sub>	Low High	V <sub>CC</sub> = Max V <sub>IN</sub> = 0.45V V <sub>IN</sub> = 5.5V			-100 40	μA
<b>Output current</b>						
I <sub>O(OFF)</sub> I <sub>OS</sub>	Hi-Z state <sup>10</sup> Short circuit <sup>4,6,7</sup>	V <sub>CC</sub> = Max V <sub>OUT</sub> = 5.5V V <sub>OUT</sub> = 0.45V V <sub>OUT</sub> = 0V	-15		80 -140 -70	μA mA
I <sub>CC</sub>	V <sub>CC</sub> supply current <sup>8</sup>	V <sub>CC</sub> = Max		250	325	mA
<b>Capacitance</b>						
I <sub>IN</sub> C <sub>B</sub>	Input I/O	V <sub>CC</sub> = 5V V <sub>IN</sub> = 2.0V V <sub>OUT</sub> = 2.0V		8 15		pF

## AC ELECTRICAL CHARACTERISTICS T<sub>A</sub> = 0°C to +75°C, 4.75 < V<sub>CC</sub> < 5.25V, R<sub>1</sub> = 470Ω, R<sub>2</sub> = 1000Ω

SYMBOL	PARAMETER		TEST CONDITIONS	LIMITS			UNIT
	TO	FROM		Min	Typ	Max	
t <sub>PD1</sub> t <sub>PD2</sub> t <sub>PD3</sub> t <sub>PD4</sub> <sup>11</sup> t <sub>PD5</sub> <sup>11</sup> t <sub>PD6</sub> <sup>11</sup> t <sub>PD7</sub> <sup>11</sup>	Output± Output± Output± Output± Output± Output±	Input± Input± Input± Input± Input± Input±	C <sub>L</sub> = 30pF			20 18 18 28 26 26	ns
	Internal					8	ns
t <sub>OE</sub> t <sub>OD</sub> <sup>9</sup>	Output- Output+	Input± Input±	C <sub>L</sub> = 5pF			20 20	ns

**NOTES:**

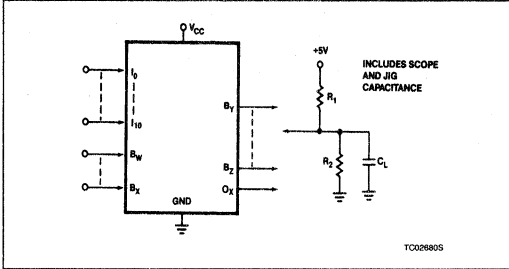
- Stresses above those listed under "Absolute Maximum Ratings" may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.
- All typical values are at V<sub>CC</sub> = 5V, T<sub>A</sub> = +25°C.
- All voltage values are with respect to network ground terminal.
- Test one at a time.
- For Pins 15-19, 21-27 and 37-40, V<sub>OL</sub> is measured with Pins 5 and 41 = 8.75V, Pin 43 = 0V and Pins 42 and 44 = 4.5V. For Pins 28-33 and 35-36, V<sub>OL</sub> is measured under same conditions EXCEPT Pin 44 = 0V.
- V<sub>OH</sub> is measured with Pins 5 and 41 = 8.75V, Pins 42 and 43 = 4.5V and Pin 44 = 0V.
- Duration of short circuit should not exceed 1 second.
- I<sub>CC</sub> is measured with all dedicated inputs at 0V and bidirectional and output pins open.
- Measured at V<sub>T</sub> = V<sub>OL</sub> + 0.5V.
- Leakage values are a combination of input and output leakage.
- Limits are guaranteed with 10 terms maximum connected to each internal feedback line.

3

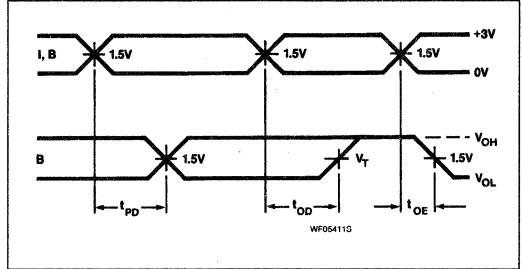
# Programmable Macro Logic Random Logic Unit\* (32 × 72 × 24)

PLHS501

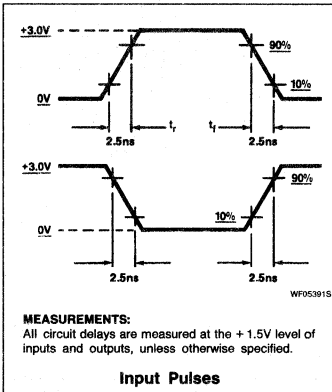
### TEST LOAD CIRCUIT



### TIMING DIAGRAM



### VOLTAGE WAVEFORM



### TIMING DEFINITIONS

- t<sub>PD1</sub> Input to Output delay, one pass, with EX-OR buffer
- t<sub>PD2</sub> Input to Output delay, one pass, with NAND buffer
- t<sub>PD3</sub> Input to Output delay, one pass, with AND buffer
- t<sub>PD4</sub> Input to Output delay, two passes, with EX-OR buffer
- t<sub>PD5</sub> Input to Output delay, two passes, with NAND buffer
- t<sub>PD6</sub> Input to Output delay, two passes, with AND buffer
- t<sub>PD7</sub> Feedback delay per internal NAND function performed
- t<sub>OD</sub> Delay between output change and when output is off (Hi-Z or High)
- t<sub>OE</sub> Delay between input change and when the output reflects specified output level

### VIRGIN STATE

A factory shipped virgin device contains all fusible links open, such that:

1. All product terms are enabled.
2. All bidirectional (B) pins are outputs.
3. All outputs are enabled.
4. All outputs are Active-High **except**  $\bar{B}_0 - \bar{B}_3$  (fusable I/O) and  $/O_4 - /O_7$  which are Active-Low.

# Programmable Macro Logic Random Logic Unit\* (32 × 72 × 24)

PLHS501

3

## PROGRAMMABLE MACRO LOGIC

The architectures of current Programmable Logic Devices (PLDs) generally employ programmable AND-OR gate strings to permit implementing logic functions directly in Sum-of-Products form (SOP). Often, that becomes a bottleneck which interferes with the efficient allocation of the on-chip logic resources. This becomes very apparent when considering very dense PLDs in high pin-count packages.

PML is a novel architecture which breaks the AND-OR bottleneck by relying instead on a single NAND-gate array to implement SOP logic functions, and to provide a central programmable "interconnect" site supporting a periphery of multi-level macros on a chip.

## INTRODUCTION

Today, system designers in a rush to bring their products to market are turning more than ever toward Programmable Logic Devices, lured by their blend of design elegance and efficiency with instant custom flexibility despite limitations in density and architecture.

To be sure, dramatic advances in Bipolar and CMOS processing technologies over the last decade have spawned denser and faster elements, with an occasional twist in architecture. Nevertheless, the structure of most current PLDs is still anchored on a fixed AND-OR gate chain linking primitive input, output, and function macros on a chip. Recently, to enhance the flexibility and design adaptability of early product offerings, denser devices have been introduced featuring logic macros with local programmable options and bypass circuitry.

However, as densities continue to increase, this approach is sure to breed devices with an overwhelming architectural maze which is altogether difficult to design, comprehend, and logically manipulate, complicating rather than simplifying their application.

PML, however, consists of various logic "access" and "process" macros coupled via a primitive, central programmable structure in the on-chip "interconnect" system (see Figure 1). Macros can have various local intelligence which, in turn, may be fixed or variable by means of programmable options also embedded in the central programmable core.

## AND-OR PLD ARCHITECTURE

From a fundamental theorem of Boolean algebra, any combinational logic function can be expressed in Sum-of-Products form involving only two levels of logic transformation. Such functions can be implemented in hard-

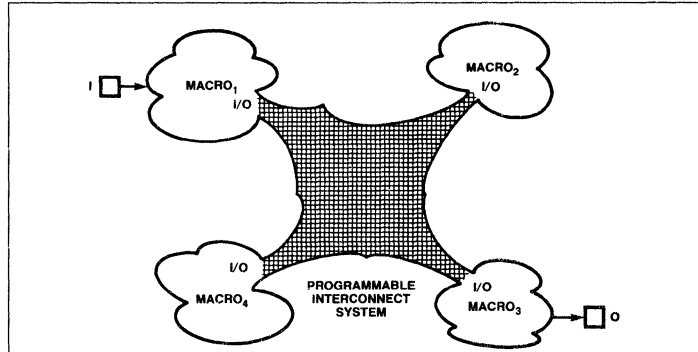


Figure 1. General PML Architecture Based on a Central Programmable "Interconnect" System Interlocking Peripheral Macros

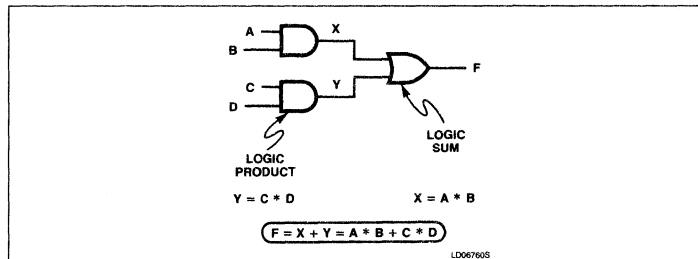


Figure 2. Typical SOP Function With Elementary AND-OR Gates

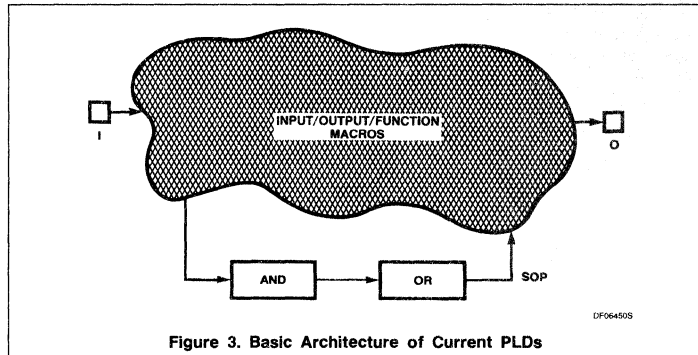


Figure 3. Basic Architecture of Current PLDs

ware by chaining two primitive logic macros as indicated in Figure 2.

The outputs of an AND-OR gate chains can be interlaced with higher level macros, such as flip-flops, to form sequential logic functions. These may be further combined into standard or custom super-macros which, in

turn, are interwoven into final system configurations. Presently, a number of PLDs are available which contain fixed sets of logic macros with programmable connections. Invariably, their architectures are anchored on a pair of concatenated AND-OR gate arrays, as diagrammed in Figure 3, to permit implementing logic functions directly in SOP form.



# Programmable Macro Logic Random Logic Unit\* (32 × 72 × 24)

PLHS501

Virtually all PLDs fall in three main categories: PROM, Fixed OR, and PLA. As illustrated in Figure 4, these differ primarily in the programmability of the AND-OR gate chain which, by being irrevocably interposed as a two-level logic link between on-chip macros and I/O paths, often can turn into a design liability.

For example, in PROMs the exponential increase in AND array size with number of inputs is a severe drawback which limits their contention in the logic application arena.

In fixed OR devices, the fixed inputs to the OR array confine their architectures to essentially a composite of AND-OR, AND-OR-register logic slices defining independent I/O paths through the chip. This causes duplication of common product terms, and waste of unused AND gates in each slice. Also, since each slice is dedicated to an output pin, devices with registered, programmable I/O pins entail an even greater waste of on-chip logic resources when configured in register-bypass or fixed input modes. Moreover, no single level logic functions can be constructed, except dedicated on-chip control functions.

Finally, even as the twin programmability of the AND-OR arrays in current PLA-type devices lends greater flexibility, and may circumvent some of the above problems, it does pose difficult process, design and performance compromises in migrating toward more dense structures.

Thus, on the whole, the AND-OR gate chain forms a bottleneck which can impair the fragmentation and reallocation of the on-chip logic resources, curtailing the adaptability and efficiency of current PLDs in applications involving unstructured algorithms and scattered logic functions.

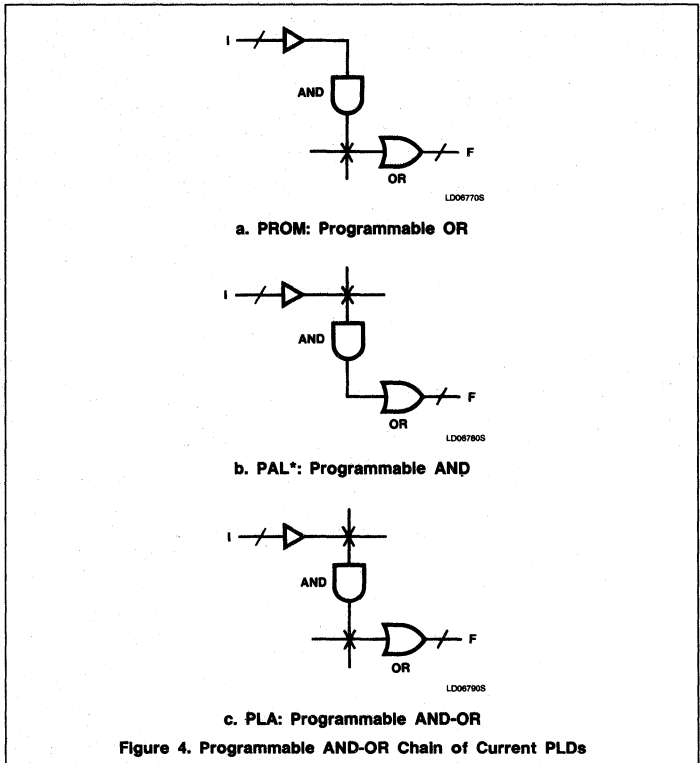


Figure 4. Programmable AND-OR Chain of Current PLDs

\*PAL is a registered trademark of Monolithic Memories, Inc.

# Programmable Macro Logic Random Logic Unit\* (32 × 72 × 24)

PLHS501

## NAND-NAND GATE EQUIVALENCE

One way of relieving the burden placed on PLDs' architecture by the AND-OR gate string is to do away with it altogether, and replace it with a new programmable structure forged out of a new compromise on silicon between "interconnect" redundancy and ease of design and application. The approach adopted for PML is based on the fundamental equivalence between two-level AND-OR and NAND-NAND gate constructs in expressing combinatorial logic functions in SOP form, as illustrated in Figure 5.

Since programmable NAND-NAND logic chains can be formed by coupling identical NAND gates with programmable inputs, the NAND-array string in Figure 6 can be implemented as a single global array with feedback.

A basic NAND-array core linking a periphery of primitive macros is shown in Figure 7. When the feedback path is just a simple wire, the NAND-array can be fragmented into single-level or double-level true/complement logic functions by programming respectively single (a-c-d), or double (a-b-c-d), interconnecting paths through the array from inputs I to output F.

More complex functions can be implemented by programming multi-level logic loops through the array.

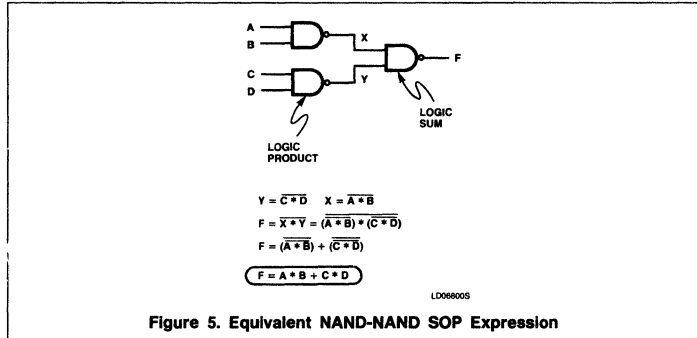


Figure 5. Equivalent NAND-NAND SOP Expression

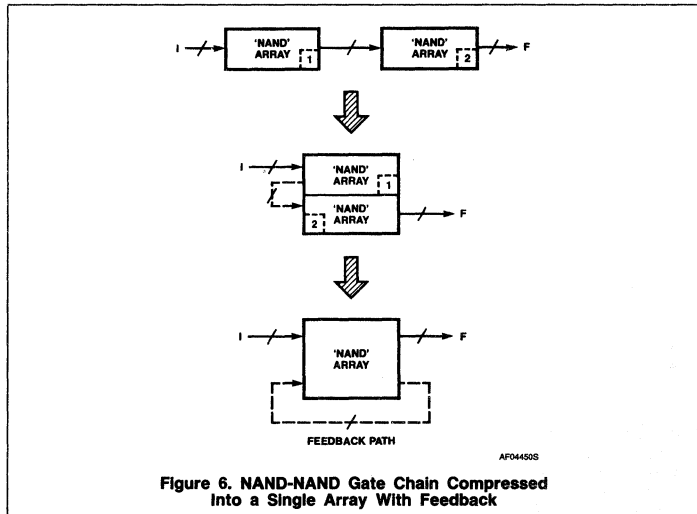


Figure 6. NAND-NAND Gate Chain Compressed into a Single Array With Feedback

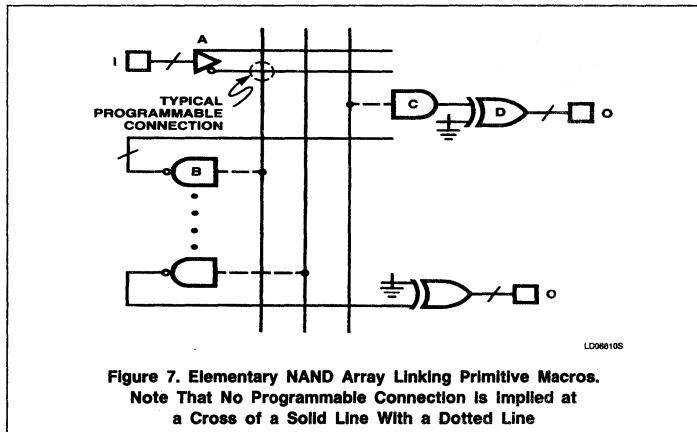


Figure 7. Elementary NAND Array Linking Primitive Macros. Note That No Programmable Connection is Implied at a Cross of a Solid Line With a Dotted Line

# Programmable Macro Logic Random Logic Unit\* (32 × 72 × 24)

PLHS501

## BASIC PML ARCHITECTURE

The primitive NAND core can be expanded to support at its periphery a structure of logic macros of growing size and complexity, including various types of flip-flops, EX-OR gates, buffers, counters, shift registers, multiplexers, decoders, arithmetic/logic units, memories, etc. As illustrated in the general structure of Figure 8, I/O macros would provide off-chip paths to the outside world, while FUNCTION macros looped about the array would provide the necessary logic processing capability.

In deference to trade-offs in functionality and speed imposed by the array size and loading, Type-1 Output (or Function) macros may be driven directly from NAND gates, while Type-2 macros may be driven by gates first looped through the array. There are virtually no limits to the number, size, and functionality of the on-chip macros surrounding the central core, other than those imposed by processing and packaging capabilities.

The main design advantage of PML is that it relies upon a single array to interconnect all macros. This reduces the design task to that of building essentially a PROM-like programmable structure more forgiving than the double array in PLA-type devices, and more flexible than that in fixed OR devices. Also, the ability to forge the NAND core into multi-level logic paths at will increases the usage efficiency of the on-chip logic resources, and greatly enhances the versatility of PML in most applications.

## THE FIRST PML OFFERING

The first PML device, designated PLHS501, is primarily a combinational logic device including only primitive macros. It is a "gate bucket" whose deceptively simple structure can implement virtually all logic functions provided by existing combinational PAL/PLA-type devices. Also, it is the first to provide true EX-OR output functions (in addition to output polarity inversion), as well as multi-level gate constructs and cross-coupled latches on-chip without involving output pins. These are shown in Figure 9.

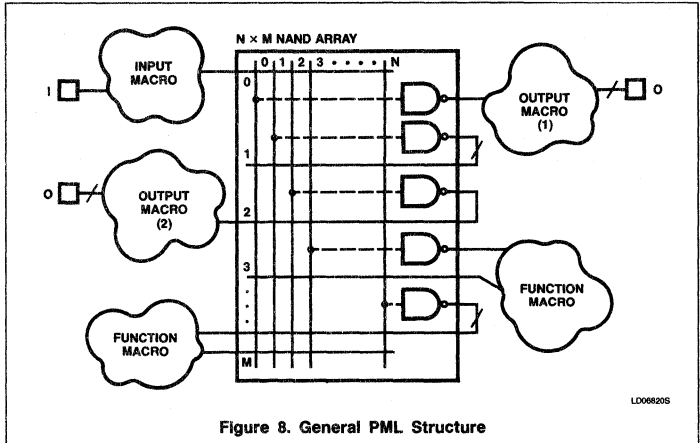


Figure 8. General PML Structure

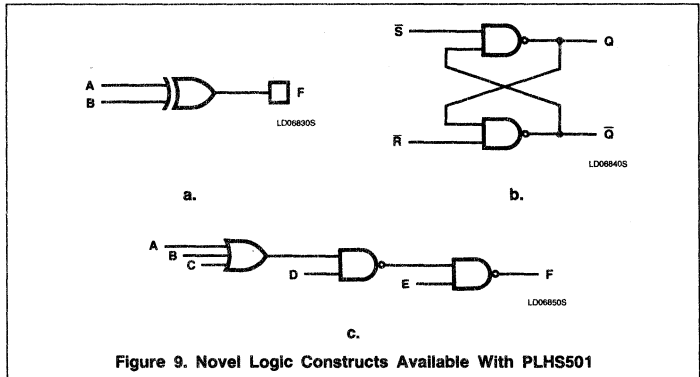


Figure 9. Novel Logic Constructs Available With PLHS501

Built with Signetics' proprietary ZA oxide isolation process, and vertical fuses, the PLHS501 has a NAND matrix about the size of a 16k-bit PROM. With a typical  $I_{CC}$  of 250mA, it features maximum input to output delays of 18/26ns over the commercial temperature range, reflecting respectively single and double passes through the NAND array. Assuming a conservative "interconnect" factor of 20%, the PLHS501 has a density

roughly equivalent to 1,500 2-input NAND gates.

The PLHS501 is the first member of a growing family of devices slated for future introduction. Built with either Bipolar or Erasable CMOS technologies, these will incorporate sets of higher level macros, leading to devices with greater logic power and a more efficient utilization of the on-chip interconnect system.

**Application Specific Products**  
● PML Series 52

*Objective Specification*

**DESCRIPTION**

The PLHS502 is a member of the Signetics Programmable Macro Logic family. PML is unique in its capability of performing other than two level logic functions without incurring I/O buffer delays. This allows the logic or system designer to imbed logical operations or macro structures such as registers within the framework of the I/O pins. Since the imbedded functions are independent of the delays created by the I/O buffers, they can be performed at speeds lesser architectures cannot reproduce.

The PLHS502 features a unique clock matrix that allows the user to individually connect each of the 16 buried registers to any of 8 different clock sources. The output  $\bar{Q}$  of the 8D type and 8 S-R type registers is internally feedback to the NAND array. In addition the output Q of each register is internally feedback to both the NAND array and the clock array. Thus the PML architecture eliminates the requirement of utilizing external I/O pins for register feedback paths. The PLHS502 is therefore ideally suited for both Synchronous and Asynchronous State Machines as well as event driven controller functions.

The technique used to perform this operation is a NAND foldback network which allows the direct interconnection of any number of logic nodes within the single fuse matrix. Macros can be formed and then interconnected to the I/O structure. In addition, single level and multiple level logic can be performed at speeds which reflect only the logic path utilized. Therefore, a single-level logic function has a very short path through the device. Additional levels incur only one NAND foldback delay per level. This delay is less than the combined delay created by previous generations of devices which stipulate that the logic signal must pass through I/O buffers after one or two levels of logic are performed.

The PLHS502 is fabricated in Signetics ZA Oxide-Isolated Bipolar Process which utilizes Vertical Avalanche Migration Programmed (VAMP™) Fuses as programming elements. These fuses provide high programming yield and reli-

ability. Proprietary onboard test circuitry allows the PLHS502 to be thoroughly tested prior to programming.

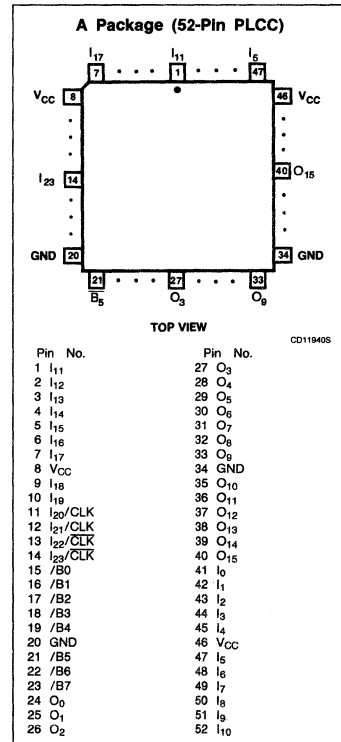
**FEATURES**

- Signetics NAND foldback architecture
- Field-Programmable
- 20 dedicated inputs
- 4 input/clock pins
- 16 buried registers
  - 8 D type
  - 8 S-R type
- 8 individual clocks
  - 4 from dedicated input/clock pins
  - 4 from NAND array
- 8 bidirectional I/O
- 16 dedicated outputs
  - 8 Active-High outputs
  - 4 outputs with polarity control
  - 4 Tri-statable outputs with polarity control
- 64 internal NAND foldback terms
- Logic delay times:
  - Single level = 22ns (max.)
  - Two logic levels = 30ns (max.)
  - Each additional internal NAND delay = 8ns (max.)
  - Maximum operating frequency  $f_{MAX} = 35\text{MHz}$
- Supported by AMAZE Development System
- Testable in unprogrammed state
- Verify Lock Fuse
- TTL compatible
- Power dissipation: 1.25W (typ.)

**APPLICATIONS**

- Any digital function requiring more than 2 levels of logic
- Systems requiring state/controller functions in conjunction with random logic
- Asynchronous state machines
- Event-driven state machines
- Logic sequencers
- Ripple counters

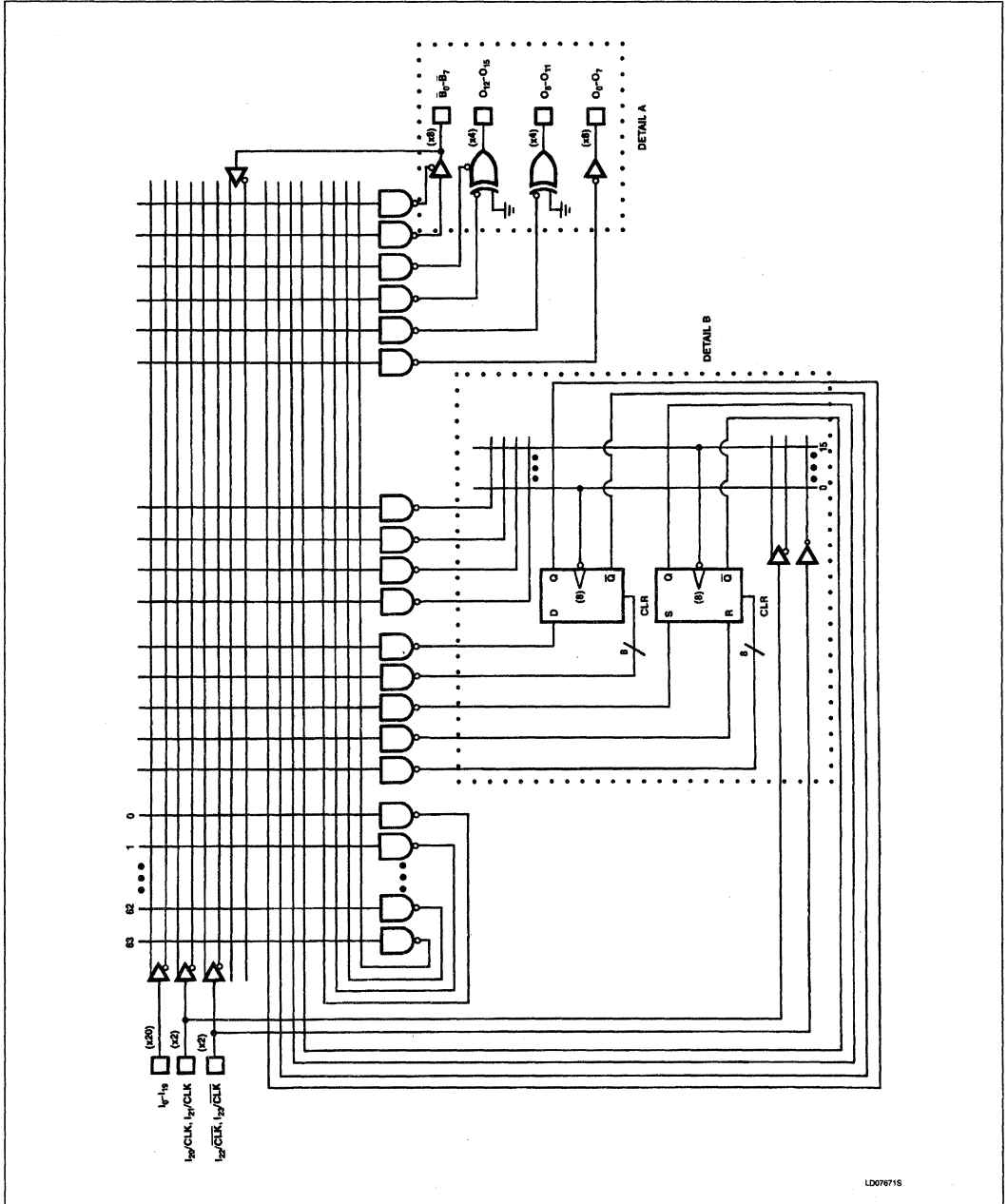
**PIN CONFIGURATION**



# Programmable Macro Logic Sync/Async Register Macro Unit (32 × 64 × 24)

PLHS502

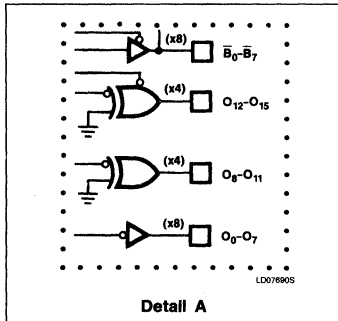
## FUNCTIONAL DIAGRAM



LD07671S

# Programmable Macro Logic Sync/Async Register Macro Unit (32 × 64 × 24)

PLHS502



## PROGRAMMABLE MACRO LOGIC

The architectures of current Programmable Logic Devices (PLDs) generally employ programmable AND-OR gate strings to permit implementing logic functions directly in Sum-of-Products form (SOP). Often, that becomes a bottleneck which interferes with the efficient allocation of the on-chip logic resources. This becomes very apparent when considering very dense PLDs in high pin-count packages.

PML is a novel architecture which breaks the AND-OR bottleneck by relying instead on a single NAND-gate array to implement SOP logic functions, and to provide a central programmable "interconnect" site supporting a periphery of multi-level macros on a chip.

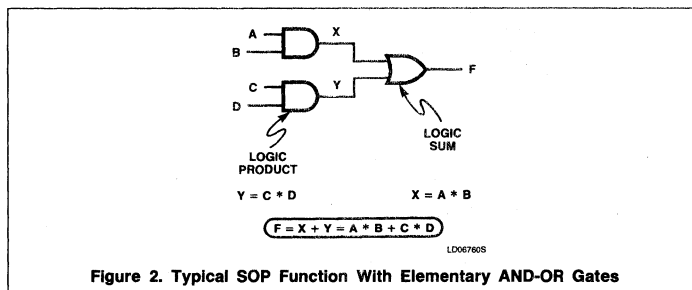
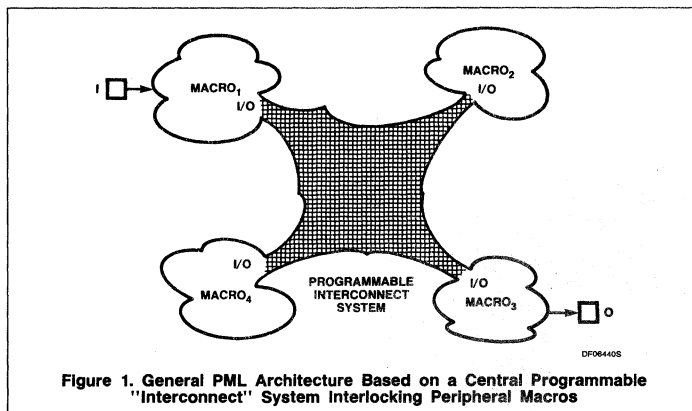
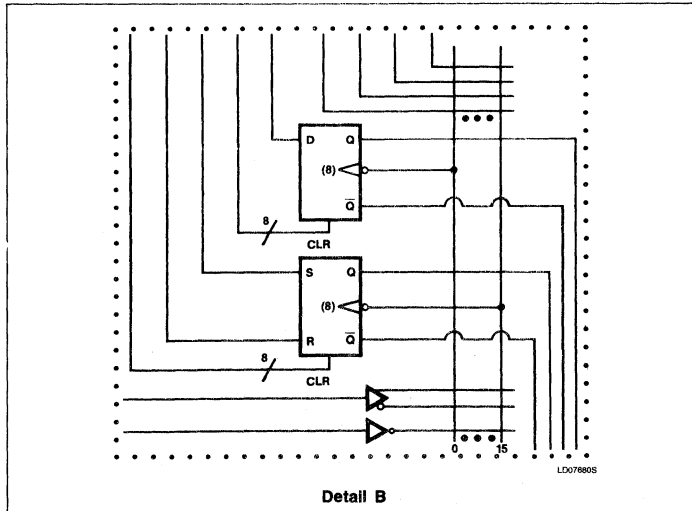
## INTRODUCTION

Today, system designers in a rush to bring their products to market are turning more than ever toward Programmable Logic Devices, lured by their blend of design elegance and efficiency with instant custom flexibility despite limitations in density and architecture.

To be sure, dramatic advances in Bipolar and CMOS processing technologies over the last decade have spawned denser and faster elements, with an occasional twist in architecture. Nevertheless, the structure of most current PLDs is still anchored on a fixed AND-OR gate chain linking primitive input, output, and function macros on a chip. Recently, to enhance the flexibility and design adaptability of early product offerings, denser devices have been introduced featuring logic macros with local programmable options and bypass circuitry.

However, as densities continue to increase, this approach is sure to breed devices with an overwhelming architectural maze which is altogether difficult to design, comprehend, and logically manipulate, complicating rather than simplifying their application.

PML, however, consists of various logic "access" and "process" macros coupled via a



primitive, central programmable structure in the on-chip "interconnect" system (see Figure 1). Macros can have various local intelli-

gence which, in turn, may be fixed or variable by means of programmable options also embedded in the central programmable core.

# Programmable Macro Logic Sync/Async Register Macro Unit (32 × 64 × 24)

PLHS502

## AND-OR PLD ARCHITECTURE

From a fundamental theorem of Boolean algebra, any combinational logic function can be expressed in Sum-of-Products form involving only two levels of logic transformation. Such functions can be implemented in hardware by chaining two primitive logic macros as indicated in Figure 2.

The outputs of an AND-OR gate chains can be interlaced with higher level macros, such as flip-flops, to form sequential logic functions. These may be further combined into standard or custom super-macros which, in turn, are interwoven into final system configurations. Presently, a number of PLDs are available which contain fixed sets of logic macros with programmable connections. Invariably, their architectures are anchored on a pair of concatenated AND-OR gate arrays, as diagrammed in Figure 3, to permit implementing logic functions directly in SOP form.

Virtually all PLDs fall in three main categories: PROM, Fixed OR, and PLA. As illustrated in Figure 4, these differ primarily in the programmability of the AND-OR gate chain which, by being irrevocably interposed as a two-level logic link between on-chip macros and I/O paths, often can turn into a design liability.

For example, in PROMs the exponential increase in AND array size with number of inputs is a severe drawback which limits their contention in the logic application arena.

In fixed OR devices, the fixed inputs to the OR array confine their architectures to essentially a composite of AND-OR, AND-OR-register logic slices defining independent I/O paths through the chip. This causes duplication of common product terms, and waste of unused AND gates in each slice. Also, since each slice is dedicated to an output pin, devices with registered, programmable I/O pins entail an even greater waste of on-chip logic resources when configured in register-bypass or fixed input modes. Moreover, no single level logic functions can be constructed, except dedicated on-chip control functions.

Finally, even as the twin programmability of the AND-OR arrays in current PLA-type devices lends greater flexibility, and may circumvent some of the above problems, it does pose difficult process, design and performance compromises in migrating toward more dense structures.

Thus, on the whole, the AND-OR gate chain forms a bottleneck which can impair the fragmentation and reallocation of the on-chip logic resources, curtailing the adaptability and efficiency of current PLDs in applications involving unstructured algorithms and scattered logic functions.

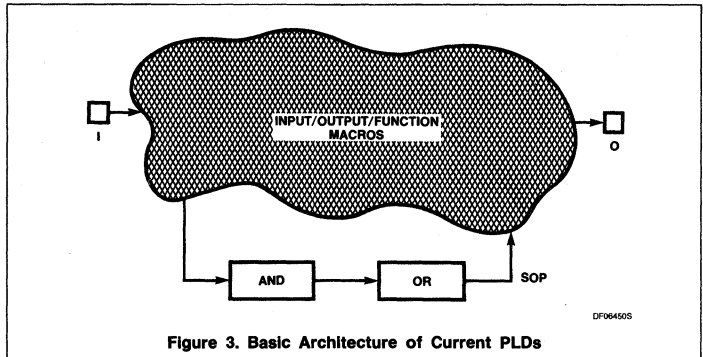


Figure 3. Basic Architecture of Current PLDs

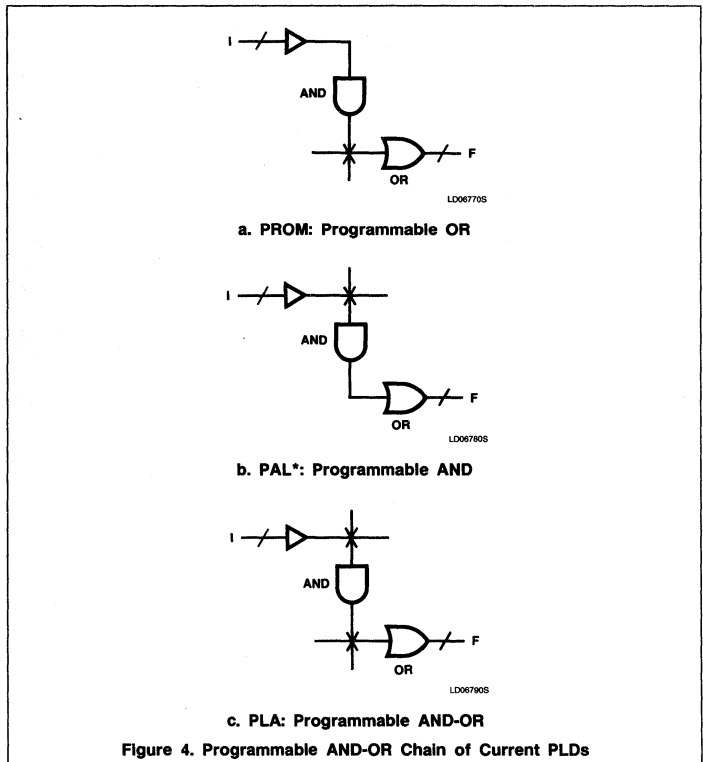


Figure 4. Programmable AND-OR Chain of Current PLDs

\*PAL is a registered trademark of Monolithic Memories, Inc.

# Programmable Macro Logic Sync/Async Register Macro Unit (32 X 64 X 24)

PLHS502

## NAND-NAND GATE EQUIVALENCE

One way of relieving the burden placed on PLDs' architecture by the AND-OR gate string is to do away with it altogether, and replace it with a new programmable structure forged out of a new compromise on silicon between "interconnect" redundancy and ease of design and application. The approach adopted for PML is based on the fundamental equivalence between two-level AND-OR and NAND-NAND gate constructs in expressing combinatorial logic functions in SOP form, as illustrated in Figure 5.

Since programmable NAND-NAND logic chains can be formed by coupling identical NAND gates with programmable inputs, the NAND-array string in Figure 6 can be implemented as a single global array with feedback.

A basic NAND-array core linking a periphery of primitive macros is shown in Figure 7. When the feedback path is just a simple wire, the NAND-array can be fragmented into single-level or double-level true/complement logic functions by programming respectively single (a-c-d), or double (a-b-c-d), interconnecting paths through the array from inputs I to output F.

More complex functions can be implemented by programming multi-level logic loops through the array.

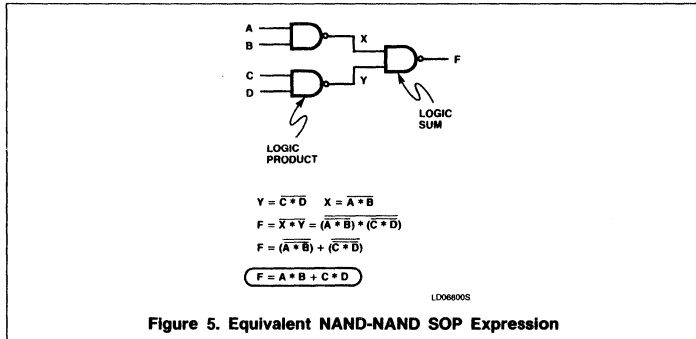


Figure 5. Equivalent NAND-NAND SOP Expression

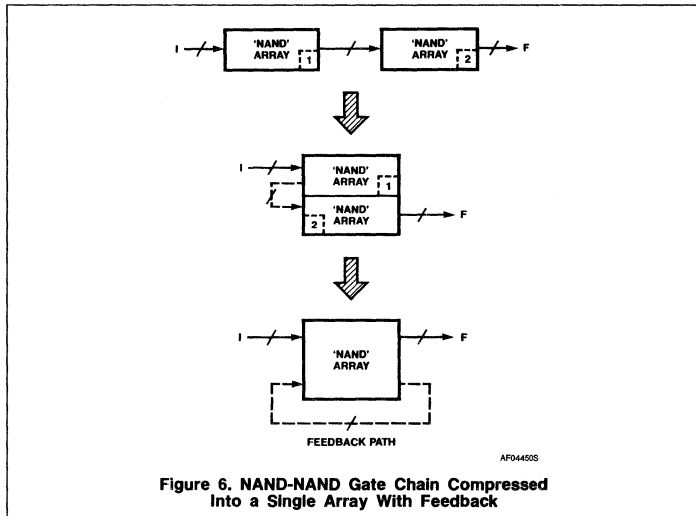


Figure 6. NAND-NAND Gate Chain Compressed Into a Single Array With Feedback

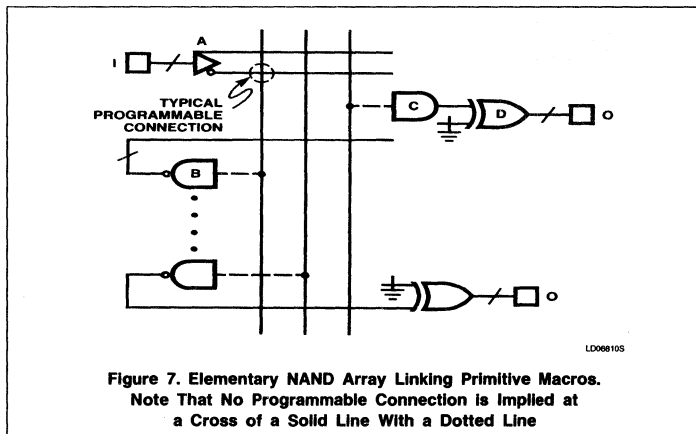


Figure 7. Elementary NAND Array Linking Primitive Macros.  
Note That No Programmable Connection is Implied at a Cross of a Solid Line With a Dotted Line



## Programmable Macro Logic Sync/Async Register Macro Unit (32 × 64 × 24)

PLHS502

### BASIC PML ARCHITECTURE

The primitive NAND core can be expanded to support at its periphery a structure of logic macros of growing size and complexity, including various types of flip-flops, EX-OR gates, buffers, counters, shift registers, multiplexers, decoders, arithmetic/logic units, memories, etc. As illustrated in the general structure of Figure 8, I/O macros would provide off-chip paths to the outside world, while FUNCTION macros looped about the array would provide the necessary logic processing capability.

In deference to trade-offs in functionality and speed imposed by the array size and loading, Type-1 Output (or Function) macros may be driven directly from NAND gates, while Type-2 macros may be driven by gates first looped through the array. There are virtually no limits to the number, size, and functionality of the on-chip macros surrounding the central core, other than those imposed by processing and packaging capabilities.

The main design advantage of PML is that it relies upon a single array to interconnect all macros. This reduces the design task to that of building essentially a PROM-like programmable structure more forgiving than the double array in PLA-type devices, and more flexible than that in fixed OR devices. Also, the ability to forge the NAND core into multi-level logic paths at will increases the usage efficiency of the on-chip logic resources, and greatly enhances the versatility of PML in most applications.

### THE FIRST PML OFFERING

The first PML device, designated PLHS501, is primarily a combinational logic device including only primitive macros. It is a "gate bucket" whose deceptively simple structure can implement virtually all logic functions provided by existing combinational PAL/PLA-type devices. Also, it is the first to provide true EX-OR output functions (in addition to output polarity inversion), as well as multi-level gate

constructs and cross-coupled latches on-chip without involving output pins. These are shown in Figure 9.

Built with Signetics' proprietary ZA oxide isolation process, and vertical fuses, the PLHS501 has a NAND matrix about the size of a 16k-bit PROM. With a typical  $I_{CC}$  of 250mA, it features maximum input to output delays of 18/26ns over the commercial temperature range, reflecting respectively single and double passes through the NAND array. Assuming a conservative "interconnect" factor of 20%, the PLHS501 has a density roughly equivalent to 1,500 2-input NAND gates.

The PLHS501 is the first member of a growing family of devices slated for future introduction. Built with either Bipolar or Erasable CMOS technologies, these will incorporate sets of higher level macros, leading to devices with greater logic power and a more efficient utilization of the on-chip interconnect system.

### INDEX

#### Section 4 – PLD Data Sheets

##### Series 20

PLS151	Field-Programmable Gate Array (18 × 15 × 12) .....	4-3
PLS153	Field-Programmable Logic Array (18 × 42 × 10) .....	4-9
PLS153A	Field-Programmable Logic Array (18 × 42 × 10) .....	4-15
PLHS153	Field-Programmable Logic Array (18 × 42 × 10) .....	4-21
PLS155	Field-Programmable Logic Sequencer (16 × 45 × 12)..	4-27
PLS157	Field-Programmable Logic Sequencer (16 × 45 × 12)..	4-37
PLS159	Field-Programmable Logic Sequencer (16 × 45 × 12)..	4-47
PLS159A	Field-Programmable Logic Sequencer (16 × 45 × 12)..	4-57
PLHS18P8A	Programmable AND Array Logic (18 × 72 × 8).....	4-67



## PLS151 Field-Programmable Gate Array (18 × 15 × 12)

Signetics Programmable Logic  
Product Specification

### Application Specific Products • Series 20

#### DESCRIPTION

The PLS151 is a single level logic element, consisting of 15 AND gates with fusible link connections for programming I/O polarity, I/O direction and output enable control.

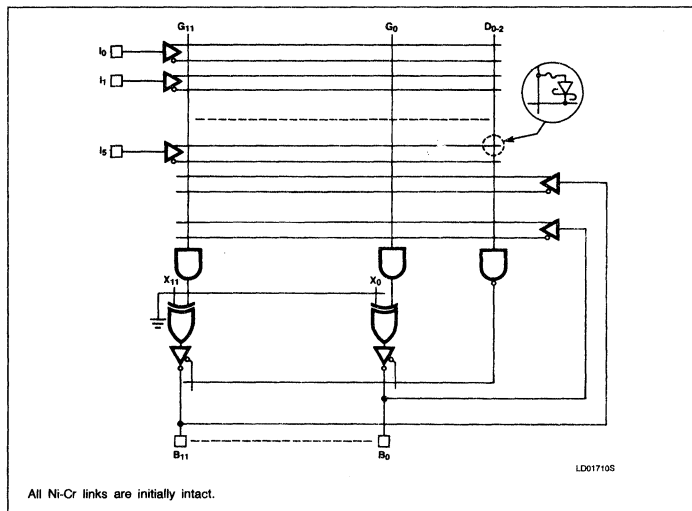
All gates are linked to 6 inputs (I) and 12 bidirectional I/O lines (B). These yield variable I/O gate configurations via 3 direction control gates (D), ranging from 18 inputs to 12 outputs.

On chip T/C buffers couple either True (I, B) or Complement ( $\bar{I}$ ,  $\bar{B}$ ) input polarities to each AND gate. The polarity of all gate outputs is individually programmable through a set of EX-OR gates for implementing AND/NAND logic functions. Alternately, if desired, OR/NOR logic functions can also be realized by programming for each gate the complement of its inputs and outputs (DeMorgan's Theorem).

The PLS151 is field programmable, enabling the user to quickly generate custom patterns using standard programming equipment.

Order codes are contained on the pages following.

#### FUNCTIONAL DIAGRAM



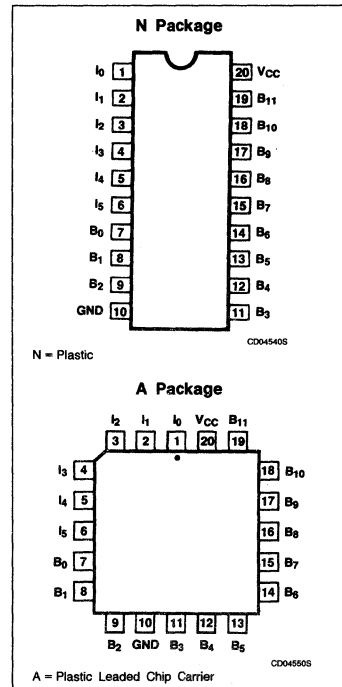
#### FEATURES

- Field-Programmable (Ni-Cr link)
- 6 inputs
- 15 product terms:
  - 12 logic terms
  - 3 control terms
- 12 bidirectional I/O lines
- Active-High or -Low outputs
- Programmable output enable
- Power dissipation: 575mW (typ.)
- I/O propagation delay: 25ns (max.)
- Input loading: -100 $\mu$ A (max.)
- TTL compatible
- Tri-state outputs

#### APPLICATIONS

- Random gating functions
- Address decoding
- Code detectors
- Memory mapped I/O
- Fault monitors
- I/O port decoders

#### PIN CONFIGURATIONS



#### LOGIC FUNCTION

##### TYPICAL OUTPUT FUNCTIONS:

###### ACTIVE-HIGH

$$X = A \cdot B \cdot C \cdot \dots$$

###### ACTIVE-LOW

$$X = \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \dots$$

$$X = \bar{A} + \bar{B} + \bar{C} + \dots$$

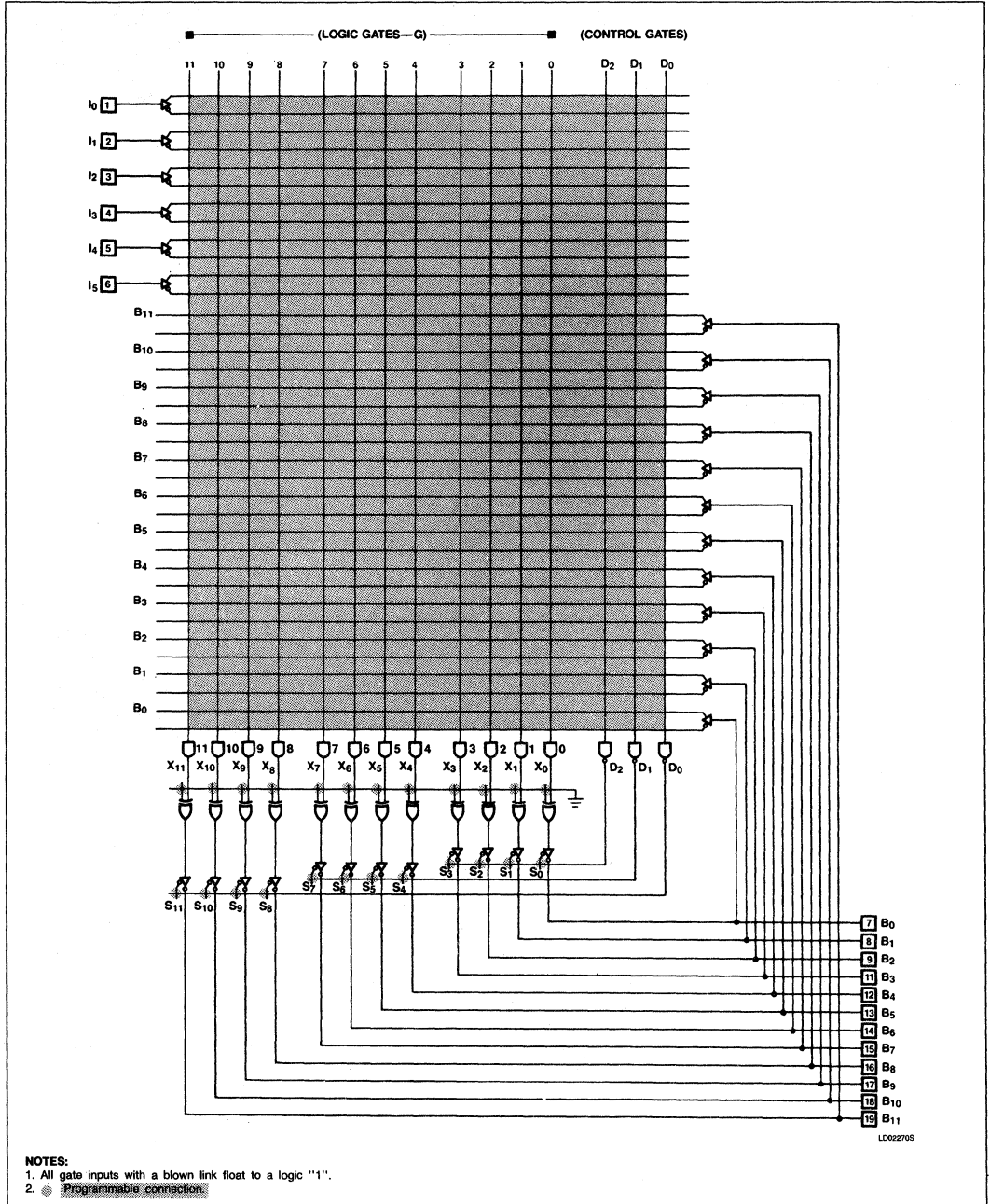
##### NOTES:

1. For each of the 12 outputs, either function X (Active-High) or  $\bar{X}$  (Active-Low) is available, but not both. The desired output polarity is programmed via the EX-OR gates.
2. X, A, B, C, etc. are user defined connections to fixed inputs (I) and bidirectional pins (B).

# Field-Programmable Gate Array (18 × 15 × 12)

PLS151

## FPGA LOGIC DIAGRAM



# Field-Programmable Gate Array (18 × 15 × 12)

PLS151

**ORDERING INFORMATION**

DESCRIPTION	ORDER CODE
20-pin Plastic DIP 300mil-wide	PLS151N
20-pin Plastic Leaded Chip Carrier	PLS151A

**THERMAL RATINGS**

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

**ABSOLUTE MAXIMUM RATINGS<sup>1</sup>**

SYMBOL	PARAMETER	RATING <sup>1</sup>		UNIT
		Min	Max	
V <sub>CC</sub>	Supply voltage		+7	V <sub>DC</sub>
V <sub>IN</sub>	Input voltage		+5.5	V <sub>DC</sub>
V <sub>OUT</sub>	Output voltage		+5.5	V <sub>DC</sub>
I <sub>IN</sub>	Input currents	-30	+30	mA
I <sub>OUT</sub>	Output currents		+100	mA
T <sub>A</sub> T <sub>STG</sub>	Temperature range Operating Storage	0 -65	+75 +150	°C

**DC ELECTRICAL CHARACTERISTICS** 0°C ≤ T<sub>A</sub> ≤ +75°C, 4.75V ≤ V<sub>CC</sub> ≤ 5.25V

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ <sup>2</sup>	Max	
<b>Input voltage<sup>3</sup></b>						
V <sub>IL</sub> V <sub>IH</sub> V <sub>IC</sub>	Low High Clamp <sup>3,4</sup>	V <sub>CC</sub> = Min V <sub>CC</sub> = Max V <sub>CC</sub> = Min, I <sub>IN</sub> = -12mA	2.0		0.8 -1.2	V
<b>Output voltage<sup>3</sup></b>						
V <sub>OL</sub> V <sub>OH</sub>	Low High	V <sub>CC</sub> = Min I <sub>OL</sub> = 10mA I <sub>OH</sub> = -2mA	2.4		0.5	V
<b>Input current<sup>7</sup></b>						
I <sub>IL</sub> I <sub>IH</sub>	Low High	V <sub>CC</sub> = Max V <sub>IN</sub> = 0.45V V <sub>IN</sub> = 5.5V			-100 40	μA
<b>Output current</b>						
I <sub>O(OFF)</sub> I <sub>OS</sub>	Hi-Z state Short circuit <sup>4,5</sup>	V <sub>CC</sub> = Max V <sub>OUT</sub> = 5.5V V <sub>OUT</sub> = 0.45V V <sub>OUT</sub> = 0V			80 -140 -70	μA mA
I <sub>CC</sub>	V <sub>CC</sub> supply current	V <sub>CC</sub> = Max		115	155	mA
<b>Capacitance</b>						
C <sub>IN</sub> C <sub>B</sub>	Input I/O	V <sub>CC</sub> = 5V V <sub>IN</sub> = 2.0V V <sub>B</sub> = 2.0V		8 15		pF

Notes on following page.

4

# Field-Programmable Gate Array (18 × 15 × 12)

PLS151

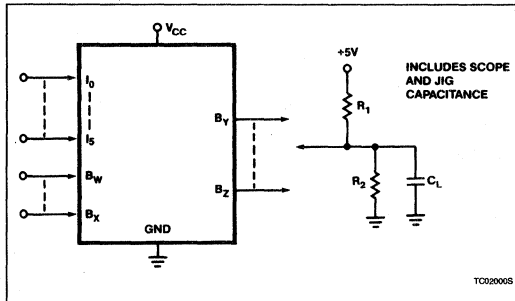
## AC ELECTRICAL CHARACTERISTICS $R_1 = 470\Omega$ , $R_2 = 1k\Omega$ , $0^\circ C \leq T_A \leq +75^\circ C$ , $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TO	FROM	TEST CONDITIONS	LIMITS			UNIT
					Min	Typ <sup>2</sup>	Max	
$t_{PD}$	Propagation delay	Output $\pm$	Input $\pm$	$C_L = 30pF$		16	25	ns
$t_{OE}$	Output enable	Output -	Input $\pm$			17	25	
$t_{OD}$	Output disable <sup>6</sup>	Output +	Input $\pm$	$C_L = 5pF$		17	25	ns

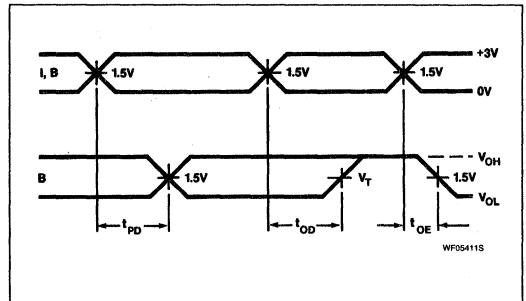
**NOTES:**

- Stresses above those listed under "Absolute Maximum Ratings" may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other conditions above those indicated in the operational and programming specification of the device is not implied.
- All typical values are at  $V_{CC} = 5V$ ,  $T_A = +25^\circ C$ .
- All voltage values are with respect to network ground terminal.
- Test one pin at a time.
- Duration of short circuit should not exceed 1 second.
- Measured at  $V_T = V_{OL} + 0.5V$ .
- $I_{IL}$  and  $I_{IH}$  limits are for dedicated inputs only ( $I_{O-}$ ).

**TEST LOAD CIRCUIT**



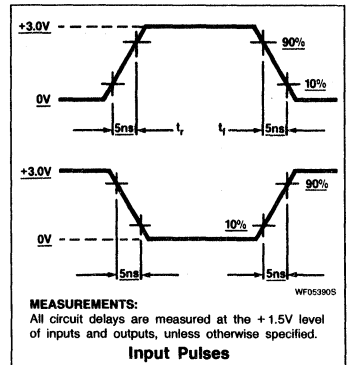
**TIMING DIAGRAM**



**TIMING DEFINITIONS**

- $t_{PD}$  Propagation delay between input and output.
- $t_{OD}$  Delay between input change and when output is off (Hi-Z or High).
- $t_{OE}$  Delay between input change and when output reflects specified output level.

**VOLTAGE WAVEFORM**



# Field-Programmable Gate Array (18 × 15 × 12)

PLS151

## LOGIC PROGRAMMING

In a virgin device all Ni-Cr links are intact. The FPGA can be programmed by means of Logic programming equipment.

With Logic programming, the AND/EX-OR gate input connections necessary to implement the desired logic function are coded directly from logic equations using the Program Table on the following page.

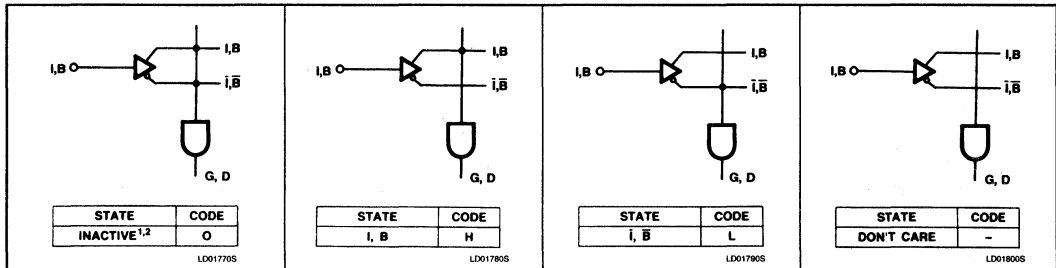
In this Table the logic state of variables I and B associated with each gate  $G_n$ ,  $D_n$  is assigned a symbol which results in the proper fusing pattern of corresponding links defined as follows:

## VIRGIN STATE

A factory shipped virgin device contains all fusible links intact, such that:

1. All outputs at 'H' polarity.
2. All  $P_n$  terms are disabled.
3. All  $P_n$  terms are active on all outputs.
4. All B terms are I/O unless otherwise programmed.

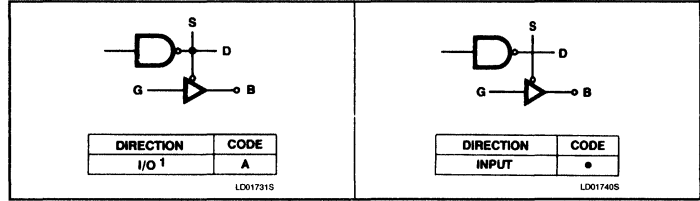
## "AND" ARRAY - (I, B)



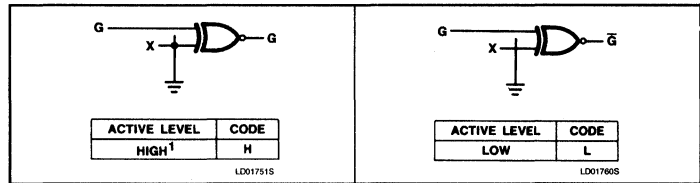
### NOTES:

1. This is the initial unprogrammed state of all links.
2. Any gate  $P_n$  will be unconditionally inhibited if both the true and complement of an input (either I or B) are left intact.

## I/O DIRECTION - (B)



## OUTPUT POLARITY - (B)



4



# Field-Programmable Gate Array (18 × 15 × 12)

PLS151

## FPGA PROGRAM TABLE

CUSTOMER NAME _____ PURCHASE ORDER # _____ SIGNETICS DEVICE # _____ CF (XXXX) CUSTOMER SYMBOLIZED PART # _____ TOTAL NUMBER OF PARTS _____ PROGRAM TABLE # _____ REV _____ DATE _____	D2 = _____ G0 = _____ G1 = _____ G2 = _____ G3 = _____ D1 = _____ G4 = _____ G5 = _____ G6 = _____ G7 = _____ D0 = _____ G8 = _____ G9 = _____ G10 = _____ G11 = _____																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
<p><b>NOTES</b></p> <ol style="list-style-type: none"> <li>1. The FPGA is shipped with all links intact. Thus a background of entries corresponding to states of virgin links exists in the table, shown BLANK for clarity.</li> <li>2. Unused I and B bits are normally programmed Don't Care (-).</li> <li>3. Unused Gates can be left blank.</li> <li>4. Disregard Polarity for Gates used only as inputs.</li> </ol>	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th rowspan="3">PIN</th> <th rowspan="3">GATE</th> <th rowspan="3">DIR</th> <th rowspan="3">POL</th> <th colspan="16">AND</th> </tr> <tr> <th colspan="8">I</th> <th colspan="8">B (I)</th> </tr> <tr> <th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th> </tr> </thead> <tbody> <tr><td>7</td><td>D2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>8</td><td>O1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>9</td><td>O2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>11</td><td>O3</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>12</td><td>D1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>13</td><td>O4</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>14</td><td>O5</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>15</td><td>O6</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>16</td><td>O7</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>17</td><td>D0</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>18</td><td>O8</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>19</td><td>O9</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>19</td><td>O10</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>19</td><td>O11</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr> <td style="padding: 5px;"> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th rowspan="2">AND</th> <th colspan="3">CONTROL</th> <th colspan="2">I/O</th> <th rowspan="2">PIN</th> <th colspan="15">VARIABLE NAME</th> </tr> <tr> <th>HIGH</th><th>H</th><th>(POL)</th> <th>A</th><th>(DIR)</th> <th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>9</th><th>8</th><th>7</th> </tr> </thead> <tbody> <tr> <td>IMACTIVE</td><td>D</td><td>H</td><td>L</td> <td></td><td></td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>I, B</td><td>H</td><td>L</td><td></td> <td></td><td></td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>I, B</td><td>L</td><td></td><td></td> <td></td><td></td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>Don't Care</td><td></td><td></td><td></td> <td></td><td></td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> </tbody> </table> </td> </tr> </tbody></table>	PIN	GATE	DIR	POL	AND																I								B (I)								5	4	3	2	1	0	11	10	9	8	7	6	5	4	3	2	1	0	7	D2																			8	O1																			9	O2																			11	O3																			12	D1																			13	O4																			14	O5																			15	O6																			16	O7																			17	D0																			18	O8																			19	O9																			19	O10																			19	O11																			<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th rowspan="2">AND</th> <th colspan="3">CONTROL</th> <th colspan="2">I/O</th> <th rowspan="2">PIN</th> <th colspan="15">VARIABLE NAME</th> </tr> <tr> <th>HIGH</th><th>H</th><th>(POL)</th> <th>A</th><th>(DIR)</th> <th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>9</th><th>8</th><th>7</th> </tr> </thead> <tbody> <tr> <td>IMACTIVE</td><td>D</td><td>H</td><td>L</td> <td></td><td></td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>I, B</td><td>H</td><td>L</td><td></td> <td></td><td></td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>I, B</td><td>L</td><td></td><td></td> <td></td><td></td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>Don't Care</td><td></td><td></td><td></td> <td></td><td></td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> </tbody> </table>	AND	CONTROL			I/O		PIN	VARIABLE NAME															HIGH	H	(POL)	A	(DIR)	6	5	4	3	2	1	19	18	17	16	15	14	13	12	11	9	8	7	IMACTIVE	D	H	L																							I, B	H	L																								I, B	L																									Don't Care																									
PIN	GATE					DIR	POL	AND																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													
								I								B (I)																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					
		5	4	3	2			1	0	11	10	9	8	7	6	5	4	3	2	1	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
7	D2																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
8	O1																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
9	O2																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
11	O3																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
12	D1																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
13	O4																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
14	O5																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
15	O6																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
16	O7																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
17	D0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
18	O8																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
19	O9																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
19	O10																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
19	O11																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th rowspan="2">AND</th> <th colspan="3">CONTROL</th> <th colspan="2">I/O</th> <th rowspan="2">PIN</th> <th colspan="15">VARIABLE NAME</th> </tr> <tr> <th>HIGH</th><th>H</th><th>(POL)</th> <th>A</th><th>(DIR)</th> <th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>9</th><th>8</th><th>7</th> </tr> </thead> <tbody> <tr> <td>IMACTIVE</td><td>D</td><td>H</td><td>L</td> <td></td><td></td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>I, B</td><td>H</td><td>L</td><td></td> <td></td><td></td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>I, B</td><td>L</td><td></td><td></td> <td></td><td></td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>Don't Care</td><td></td><td></td><td></td> <td></td><td></td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> </tbody> </table>	AND	CONTROL			I/O		PIN	VARIABLE NAME															HIGH	H	(POL)	A	(DIR)	6	5	4	3	2	1	19	18	17	16	15	14	13	12	11	9	8	7	IMACTIVE	D	H	L																							I, B	H	L																								I, B	L																									Don't Care																																																																																																																																																																																																																																																																																																																																																																									
AND		CONTROL			I/O			PIN	VARIABLE NAME																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
	HIGH	H	(POL)	A	(DIR)	6	5		4	3	2	1	19	18	17	16	15	14	13	12	11	9	8	7																																																																																																																																																																																																																																																																																																																																																																																																																																																																													
IMACTIVE	D	H	L																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		
I, B	H	L																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
I, B	L																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																				
Don't Care																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					

TB017208

## PLS153 Field-Programmable Logic Array (18 × 42 × 10)

Signetics Programmable Logic  
Product Specification

### Application Specific Products • Series 20

#### DESCRIPTION

The PLS153 is a two-level logic element, consisting of 42 AND gates and 10 OR gates with fusible link connections for programming I/O polarity and direction.

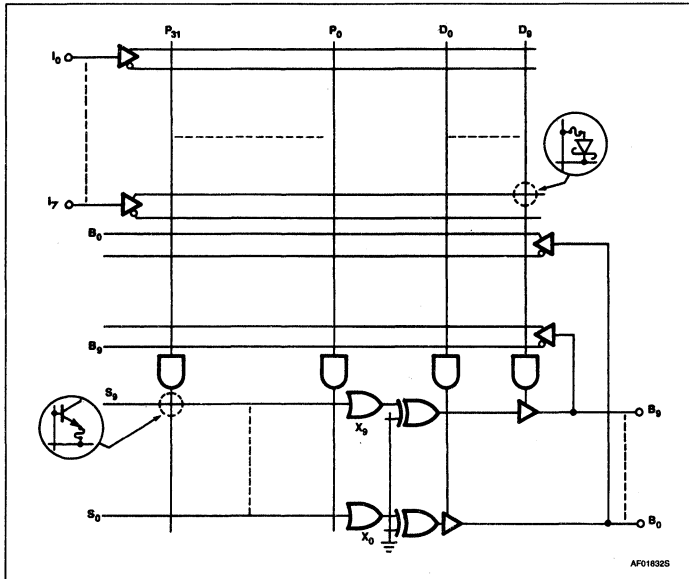
All AND gates are linked to 8 inputs (I) and 10 bidirectional I/O lines (B). These yield variable I/O gate configurations via 10 direction control gates (D), ranging from 18 inputs to 10 outputs.

On chip T/C buffers couple either True (I, B) or Complement ( $\bar{I}$ ,  $\bar{B}$ ) input polarities to all AND gates, whose outputs can be optionally linked to all OR gates. Their output polarity, in turn, is individually programmable through a set of EX-OR gates for implementing AND/OR or AND/NOR logic functions.

The PLS153 is field programmable, enabling the user to quickly generate custom patterns using standard programming equipment.

Order codes are contained on the pages following.

#### FUNCTIONAL DIAGRAM



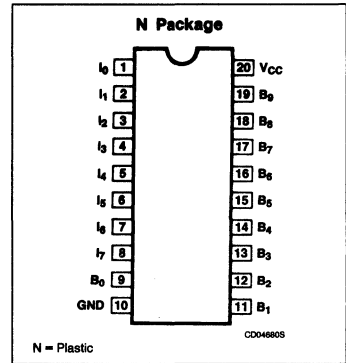
#### FEATURES

- Field-Programmable (Ni-Cr links)
- 8 Inputs
- 42 AND gates
- 10 OR gates
- 10 bidirectional I/O lines
- Active-High or -Low outputs
- 42 product terms:
  - 32 logic terms
  - 10 control terms
- I/O propagation delay: 40ns (max.)
- Input loading: -100 $\mu$ A (max.)
- Power dissipation: 650mW (typ.)
- Tri-state outputs
- TTL compatible

#### APPLICATIONS

- Random logic
- Code converters
- Fault detectors
- Function generators
- Address mapping
- Multiplexing

#### PIN CONFIGURATION



#### LOGIC FUNCTION

**TYPICAL PRODUCT TERM:**  
 $P_n = A \cdot B \cdot C \cdot D \dots$

**TYPICAL LOGIC FUNCTION:**  
**AT OUTPUT POLARITY = H**  
 $Z = P_0 + P_1 + P_2 \dots$

**AT OUTPUT POLARITY = L**  
 $Z = \bar{P}_0 \cdot \bar{P}_1 \cdot \bar{P}_2 \dots$

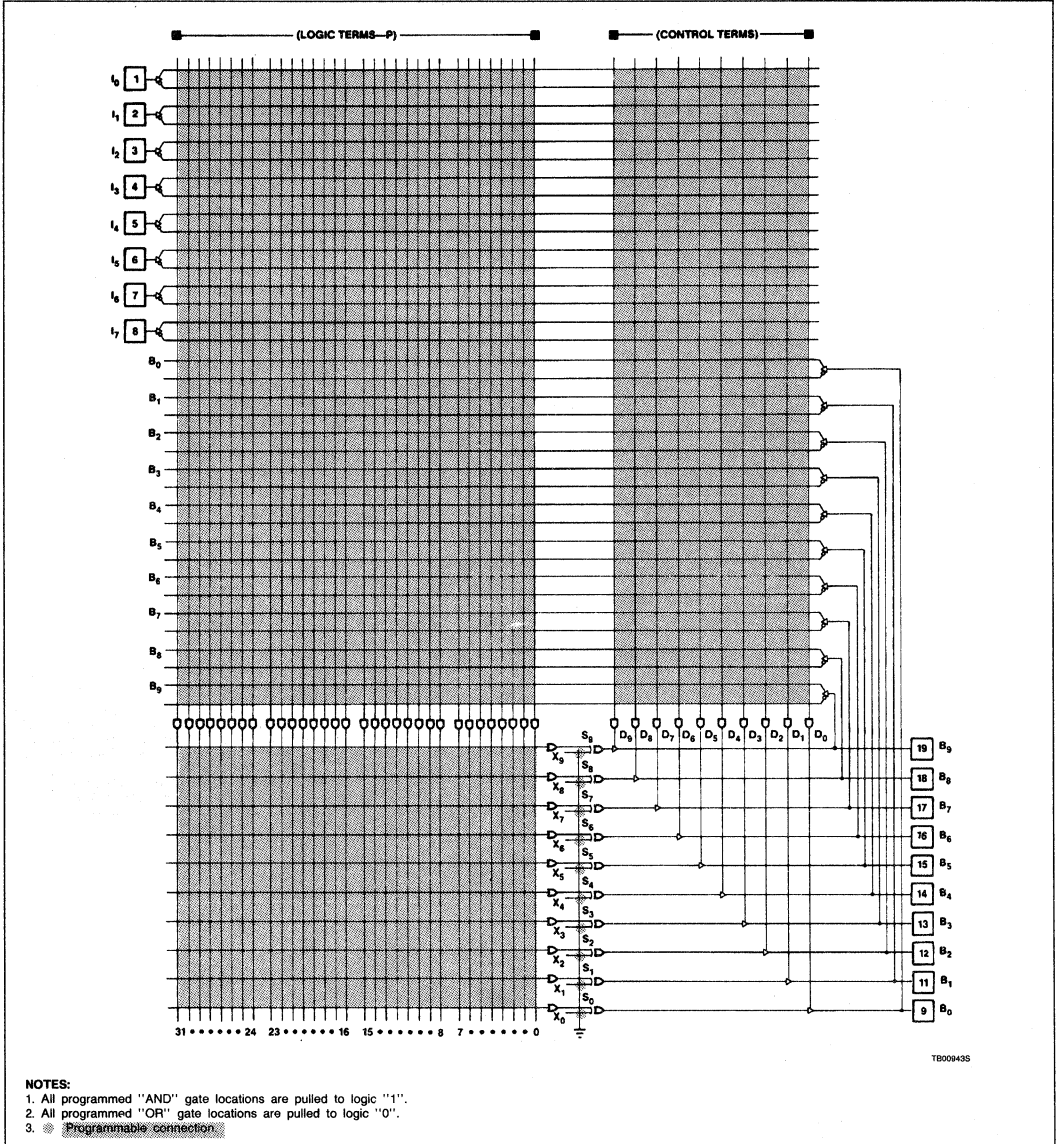
**NOTES:**

1. For each of the 10 outputs, either function Z (Active-High) or  $\bar{Z}$  (Active-Low) is available, but not both. The desired output polarity is programmed via the EX-OR gates.
2. Z, A, B, C, etc. are user defined connections to fixed inputs (I) and bidirectional pins (B).

# Field-Programmable Logic Array (18 × 42 × 10)

PLS153

## FPLA LOGIC DIAGRAM



# Field-Programmable Logic Array (18 × 42 × 10)

# PLS153

## ORDERING INFORMATION

DESCRIPTION	ORDER CODE
20-pin Plastic DIP 300mil-wide	PLS153N

## ABSOLUTE MAXIMUM RATINGS<sup>1</sup>

SYMBOL	PARAMETER	RATING		UNIT
		Min	Max	
V <sub>CC</sub>	Supply voltage		+7	V <sub>DC</sub>
V <sub>IN</sub>	Input voltage		+5.5	V <sub>DC</sub>
V <sub>OUT</sub>	Output voltage		+5.5	V <sub>DC</sub>
I <sub>IN</sub>	Input currents	-30	+30	mA
I <sub>OUT</sub>	Output currents		+100	mA
T <sub>A</sub> T <sub>STG</sub>	Temperature range Operating Storage	0 -65	+75 +150	°C

## THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

The PLS153 device is also processed to military requirements for operation over the military temperature range. For specifications and ordering information consult the Signetics Military Data Book.

4

## DC ELECTRICAL CHARACTERISTICS 0°C ≤ T<sub>A</sub> ≤ 75°C, 4.75V ≤ V<sub>CC</sub> ≤ 5.25V

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ <sup>2</sup>	Max	
<b>Input voltage<sup>3</sup></b>						
V <sub>IL</sub> V <sub>IH</sub> V <sub>IC</sub>	Low High Clamp <sup>3,4</sup>	V <sub>CC</sub> = Min V <sub>CC</sub> = Max V <sub>CC</sub> = Min, I <sub>IN</sub> = -12mA	2.0		0.8 -1.2	V
<b>Output voltage</b>						
V <sub>OL</sub> V <sub>OH</sub>	Low <sup>3,5</sup> High <sup>3,6</sup>	V <sub>CC</sub> = Min I <sub>OL</sub> = 15mA I <sub>OH</sub> = -2mA	2.4		0.5	V
<b>Input current<sup>11</sup></b>						
I <sub>IL</sub> I <sub>IH</sub>	Low High	V <sub>CC</sub> = Max V <sub>IN</sub> = 0.45V V <sub>IN</sub> = 5.5V			-100 40	μA
<b>Output current</b>						
I <sub>O(OFF)</sub> I <sub>OS</sub>	Hi-Z state <sup>10</sup> Short circuit <sup>4,6,7</sup>	V <sub>CC</sub> = Max V <sub>OUT</sub> = 5.5V V <sub>OUT</sub> = 0.45V V <sub>OUT</sub> = 0V			80 -140 -70	μA mA
I <sub>CC</sub>	V <sub>CC</sub> supply current <sup>8</sup>	V <sub>CC</sub> = Max		130	155	mA
<b>Capacitance</b>						
C <sub>IN</sub> C <sub>B</sub>	Input I/O	V <sub>CC</sub> = 5V V <sub>IN</sub> = 2.0V V <sub>B</sub> = 2.0V		8 15		pF

Notes on following page.

# Field-Programmable Logic Array (18 × 42 × 10)

PLS153

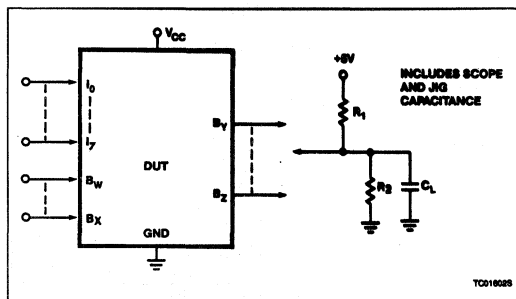
## AC ELECTRICAL CHARACTERISTICS $R_1 = 470\Omega$ , $R_2 = 1k\Omega$ , $0^\circ C \leq T_A \leq +75^\circ C$ , $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TO	FROM	TEST CONDITIONS	LIMITS			UNIT
					Min	Typ <sup>2</sup>	Max	
$t_{PD}$	Propagation delay	Output±	Input±	$C_L = 30pF$		30	40	ns
$t_{OE}$	Output enable	Output-	Input±	$C_L = 30pF$		25	35	ns
$t_{OD}$	Output disable <sup>9</sup>	Output+	Input±	$C_L = 5pF$		25	35	ns

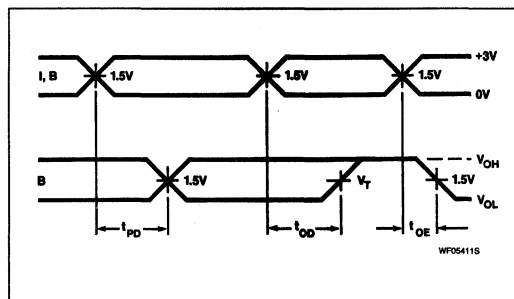
**NOTES:**

- Stresses above those listed under "Absolute Maximum Ratings" may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other conditions above those indicated in the operational and programming specification of the device is not implied.
- All typical values are at  $V_{CC} = 5V$ ,  $T_A = +25^\circ C$ .
- All voltage values are with respect to network ground terminal.
- Test one at a time.
- Measured with +10V applied to  $I_T$ .
- Measured with +10V applied to  $I_0-7$ . Output sink current is supplied through a resistor to  $V_{CC}$ .
- Duration of short circuit should not exceed 1 second.
- $I_{CC}$  is measured with  $I_0$ ,  $I_1$  at 0V,  $I_2-I_7$  and  $B_0-9$  at 4.5V.
- Measured at  $V_T = V_{OL} + 0.5V$ .
- Leakage values are a combination of input and output leakage.
- $I_{IL}$  and  $I_{IH}$  limits are for dedicated inputs only ( $I_0-I_7$ ).

### TEST LOAD CIRCUIT



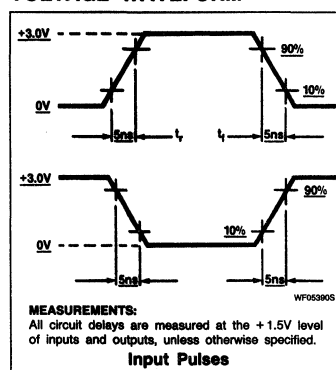
### TIMING DIAGRAM



### TIMING DEFINITIONS

- $t_{PD}$  Propagation delay between input and output.
- $t_{OD}$  Delay between input change and when output is off (Hi-Z or High).
- $t_{OE}$  Delay between input change and when output reflects specified output level.

### VOLTAGE WAVEFORM



# Field-Programmable Logic Array (18 × 42 × 10)

PLS153

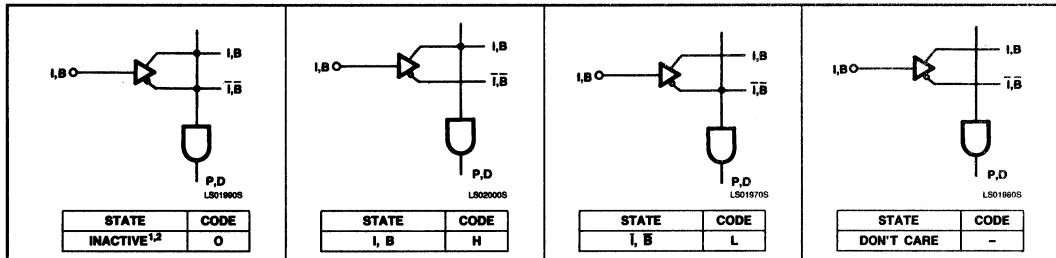
## LOGIC PROGRAMMING

The FPLA can be programmed by means of Logic programming equipment.

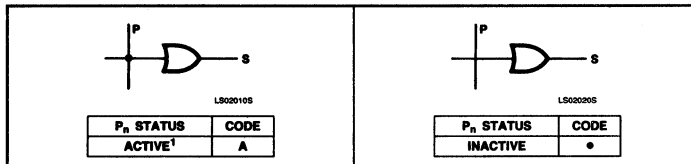
With Logic programming, the AND/OR/EX-OR gate input connections necessary to implement the desired logic function are coded directly from logic equations using the Program Table on the following page.

In this Table the logic state of variables I, P, and B associated with each Sum Term S is assigned a symbol which results in the proper fusing pattern of corresponding links defined as follows:

### "AND" ARRAY - (I, B)



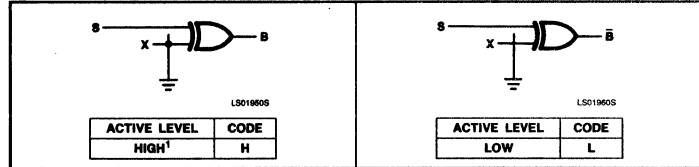
### OR ARRAY - (B)



#### NOTES:

1. This is the initial unprogrammed state of all links.
2. Any gate P<sub>n</sub> will be unconditionally inhibited if both the true and complement of an input (either I or B) are left intact.

### OUTPUT POLARITY - (B)



### VIRGIN STATE

A factory shipped virgin device contains all fusible links intact, such that:

1. All outputs at "H" polarity.
2. All P<sub>n</sub> terms are disabled.
3. All P<sub>n</sub> terms are active on all outputs.

4

# Field-Programmable Logic Array (18 × 42 × 10)

PLS153

## FPLA PROGRAM TABLE

AND		OR		POLARITY		AND																		OR																																																	
						I									B(I)									B(0)									B(0)																																								
						7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0																														
Y	E	M	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	PH	8	7	6	5	4	3	2	1	19	18	17	16	15	14	13	12	11	9	19	18	17	16	15	14	13	12	11	9
CUSTOMER NAME _____		PURCHASE ORDER # _____		SIGNETICS DEVICE # _____		CUSTOMER SYMBOLIZED PART # _____		TOTAL NUMBER OF PARTS _____		PROGRAM TABLE # _____		REV _____		DATE _____																																																											
Notes																																																																									
1. The FPLA is shipped with all links intact. Thus a background of (Shown BLANK for clarity) 2. Unused I and B bits in the AND array must be programmed Don't Care (-). 3. Unused product terms can be left blank.																																																																									
AND: <input type="checkbox"/> INACTIVE, <input type="checkbox"/> H, <input type="checkbox"/> L, <input type="checkbox"/> DONT CARE, <input type="checkbox"/> - OR: <input type="checkbox"/> ACTIVE, <input type="checkbox"/> B(0), <input type="checkbox"/> INACTIVE, <input type="checkbox"/> - CONTROL: <input type="checkbox"/> HIGH, <input type="checkbox"/> LOW, <input type="checkbox"/> I, <input type="checkbox"/> L, <input type="checkbox"/> POL																																																																									
VARIABLE NAME																																																																									

TB009555

## PLS153A Field-Programmable Logic Array (18 × 42 × 10)

Signetics Programmable Logic  
Product Specification

### Application Specific Products • Series 20

#### DESCRIPTION

The PLS153A is a two-level logic elements, consisting of 42 AND gates and 10 OR gates with fusible link connections for programming I/O polarity and direction.

All AND gates are linked to 8 inputs (I) and 10 bidirectional I/O lines (B). These yield variable I/O gate configurations via 10 direction control gates (D), ranging from 18 inputs to 10 outputs.

On chip T/C buffers couple either True (I, B) or Complement ( $\bar{I}$ ,  $\bar{B}$ ) input polarities to all AND gates, whose outputs can be optionally linked to all OR gates. Their output polarity, in turn, is individually programmable through a set of EX-OR gates for implementing AND/OR or AND/NOR logic functions.

The PLS153A is field programmable, enabling the user to quickly generate custom patterns using standard programming equipment.

Order codes are contained on the pages following.

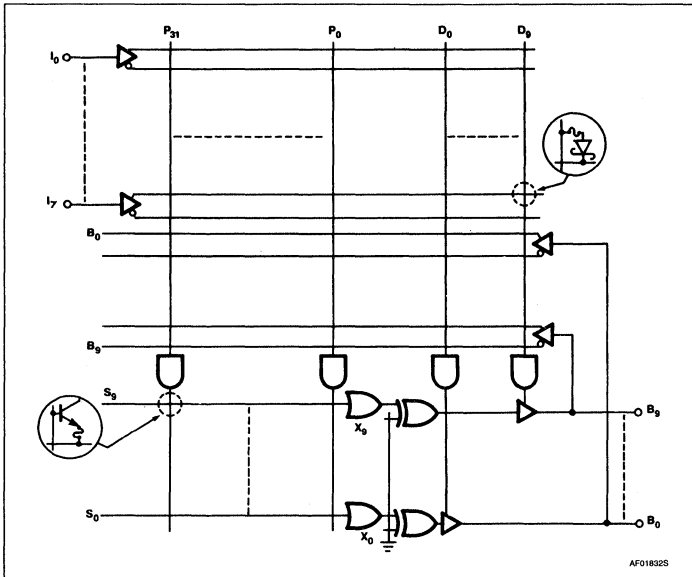
#### FEATURES

- Field-Programmable (Ni-Cr links)
- 8 Inputs
- 42 AND gates
- 10 OR gates
- 10 bidirectional I/O lines
- Active-High or -Low outputs
- 42 product terms:
  - 32 logic terms
  - 10 control terms
- I/O propagation delay: 30ns (max.)
- Input loading: -100 $\mu$ A (max.)
- Power dissipation: 650mW (typ.)
- Tri-state outputs
- TTL compatible

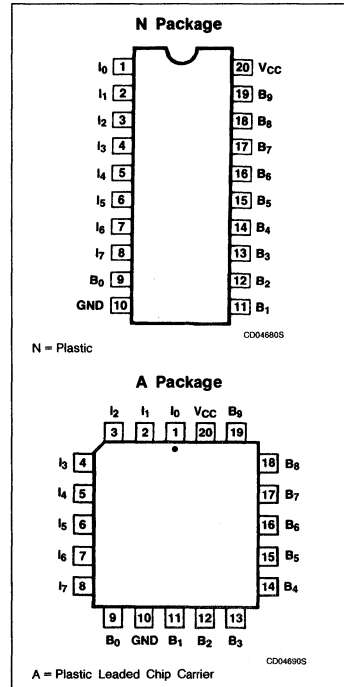
#### APPLICATIONS

- Random logic
- Code converters
- Fault detectors
- Function generators
- Address mapping
- Multiplexing

#### FUNCTIONAL DIAGRAM



#### PIN CONFIGURATIONS



#### LOGIC FUNCTION

**TYPICAL PRODUCT TERM:**  
 $P_n = A \cdot B \cdot C \cdot D \dots$   
**TYPICAL LOGIC FUNCTION:**  
**AT OUTPUT POLARITY = H**  
 $Z = P_0 + P_1 + P_2 \dots$   
**AT OUTPUT POLARITY = L**  
 $Z = \bar{P}_0 + \bar{P}_1 + \bar{P}_2 \dots$   
 $Z = \bar{P}_0 \cdot \bar{P}_1 \cdot \bar{P}_2 \dots$

#### NOTES:

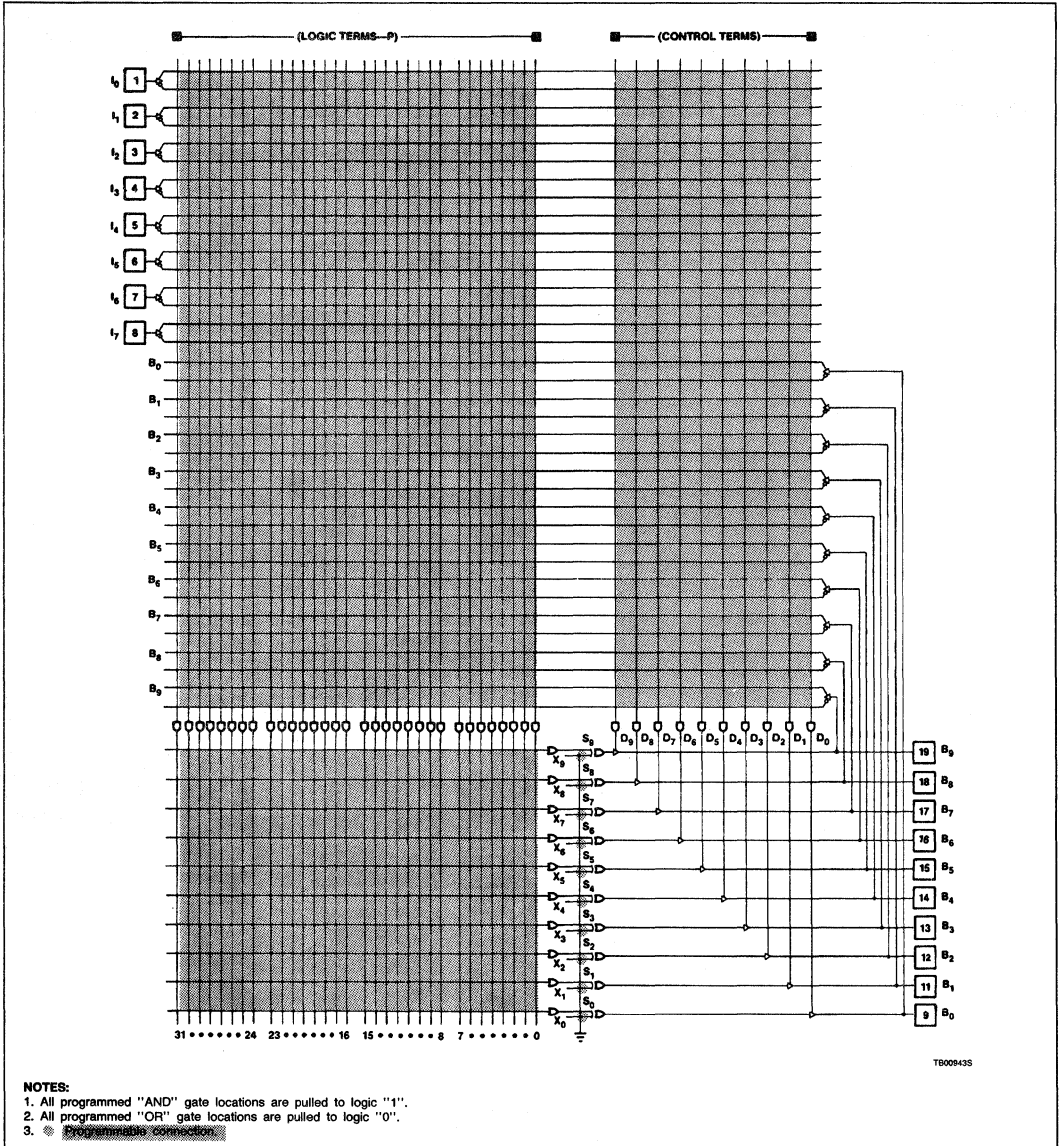
1. For each of the 10 outputs, either function Z (Active-High) or  $\bar{Z}$  (Active-Low) is available, but not both. The desired output polarity is programmed via the EX-OR gates.
2. Z, A, B, C, etc. are user defined connections to fixed inputs (I) and bidirectional pins (B).



# Field-Programmable Logic Array (18 × 42 × 10)

PLS153A

## FPLA LOGIC DIAGRAM



# Field-Programmable Logic Array (18 × 42 × 10)

PLS153A

## ORDERING INFORMATION

DESCRIPTION	ORDER CODE
20-pin Plastic DIP 300mil-wide	PLS153AN
20-pin Plastic Leaded Chip Carrier	PLS153AA

## THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

## ABSOLUTE MAXIMUM RATINGS<sup>1</sup>

SYMBOL	PARAMETER	RATING		UNIT
		Min	Max	
V <sub>CC</sub>	Supply voltage		+7	V <sub>DC</sub>
V <sub>IN</sub>	Input voltage		+5.5	V <sub>DC</sub>
V <sub>OUT</sub>	Output voltage		+5.5	V <sub>DC</sub>
I <sub>IN</sub>	Input currents	-30	+30	mA
I <sub>OUT</sub>	Output currents		+100	mA
T <sub>A</sub> T <sub>STG</sub>	Temperature range Operating Storage	0 -65	+75 +150	°C

## DC ELECTRICAL CHARACTERISTICS 0°C ≤ T<sub>A</sub> ≤ 75°C, 4.75V ≤ V<sub>CC</sub> ≤ 5.25V

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ <sup>2</sup>	Max	
<b>Input voltage<sup>3</sup></b>						
V <sub>IL</sub> V <sub>IH</sub> V <sub>IC</sub>	Low High Clamp <sup>3,4</sup>	V <sub>CC</sub> = Min V <sub>CC</sub> = Max V <sub>CC</sub> = Min, I <sub>IN</sub> = -12mA	2.0	-0.8	0.8 -1.2	V
<b>Output voltage</b>						
V <sub>OL</sub> V <sub>OH</sub>	Low <sup>3,5</sup> High <sup>3,6</sup>	V <sub>CC</sub> = Min I <sub>OL</sub> = 15mA I <sub>OH</sub> = -2mA	2.4		0.5	V
<b>Input current<sup>11</sup></b>						
I <sub>IL</sub> I <sub>IH</sub>	Low High	V <sub>CC</sub> = Max V <sub>IN</sub> = 0.45V V <sub>IN</sub> = 5.5V			-100 40	μA
<b>Output current</b>						
I <sub>O(OFF)</sub> I <sub>OS</sub>	Hi-Z state <sup>10</sup> Short circuit <sup>4,6,7</sup>	V <sub>CC</sub> = Max V <sub>OUT</sub> = 5.5V V <sub>OUT</sub> = 0.45V V <sub>OUT</sub> = 0V	-15		80 -140 -70	μA mA
I <sub>CC</sub>	V <sub>CC</sub> supply current <sup>8</sup>	V <sub>CC</sub> = Max		130	155	mA
<b>Capacitance</b>						
C <sub>IN</sub> C <sub>B</sub>	Input I/O	V <sub>CC</sub> = 5V V <sub>IN</sub> = 2.0V V <sub>B</sub> = 2.0V		8 15		pF

Notes on following page.

4

# Field-Programmable Logic Array (18 × 42 × 10)

PLS153A

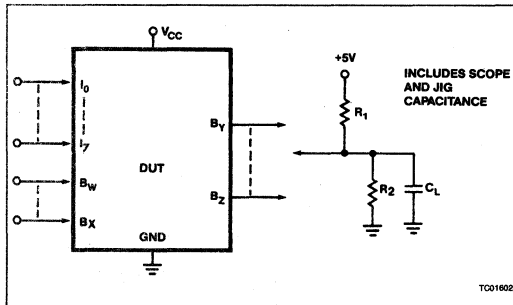
## AC ELECTRICAL CHARACTERISTICS $R_1 = 470\Omega$ , $R_2 = 1k\Omega$ , $0^\circ C \leq T_A \leq +75^\circ C$ , $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TO	FROM	TEST CONDITIONS	LIMITS			UNIT
					Min	Typ <sup>2</sup>	Max	
$t_{PD}$	Propagation delay	Output±	Input±	$C_L = 30pF$		20	30	ns
$t_{OE}$	Output enable	Output±	Input±	$C_L = 30pF$		20	30	ns
$t_{OD}$	Output disable <sup>9</sup>	Output±	Input±	$C_L = 5pF$		20	30	ns

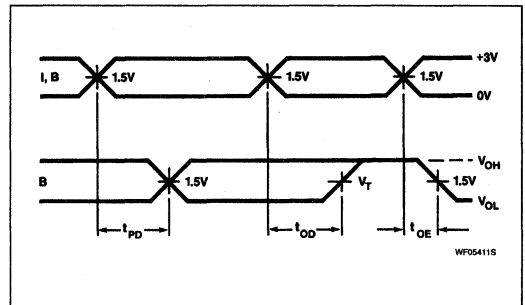
**NOTES:**

- Stresses above those listed under "Absolute Maximum Ratings" may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other conditions above those indicated in the operational and programming specification of the device is not implied.
- All typical values are at  $V_{CC} = 5V$ ,  $T_A = +25^\circ C$ .
- All voltage values are with respect to network ground terminal.
- Test one at a time.
- Measured with +10V applied to  $I_7$ .
- Measured with +10V applied to  $I_{0-7}$ . Output sink current is supplied through a resistor to  $V_{CC}$ .
- Duration of short circuit should not exceed 1 second.
- $I_{CC}$  is measured with  $I_{0,1}$  at 0V and  $I_{2-7}$  and  $B_{0-9}$  at 4.5V.
- Measured at  $V_T = V_{OL} + 0.5V$ .
- Leakage values are a combination of input and output leakage.
- $I_{IH}$  and  $I_{IL}$  limits are for dedicated inputs only ( $I_0-I_9$ ).

### TEST LOAD CIRCUIT



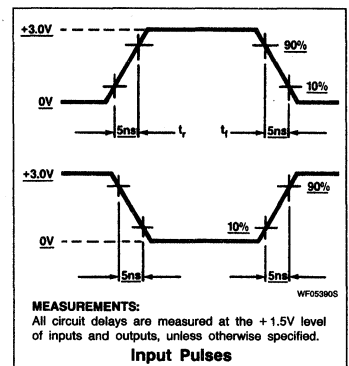
### TIMING DIAGRAM



### TIMING DEFINITIONS

- $t_{PD}$  Propagation delay between input and output.
- $t_{OD}$  Delay between input change and when output is off (Hi-Z or High).
- $t_{OE}$  Delay between input change and when output reflects specified output level.

### VOLTAGE WAVEFORM



# Field-Programmable Logic Array (18 × 42 × 10)

PLS153A

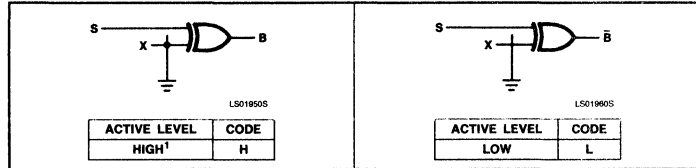
## LOGIC PROGRAMMING

The FPLA can be programmed by means of Logic programming equipment.

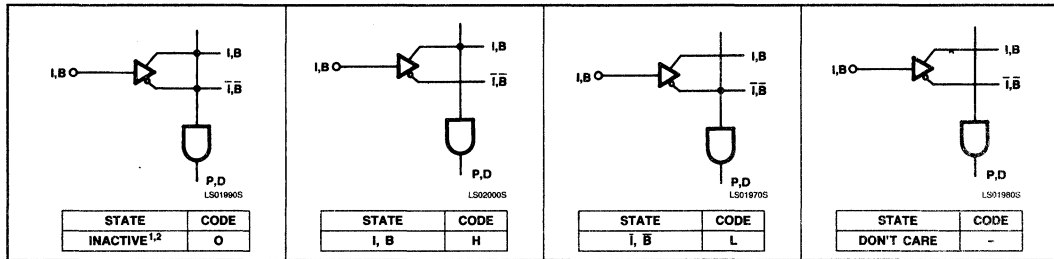
With Logic programming, the AND/OR/EX-OR gate input connections necessary to implement the desired logic function are coded directly from logic equations using the Program Table on the following page.

In this Table the logic state of variables I, P, and B associated with each Sum Term S is assigned a symbol which results in the proper fusing pattern of corresponding links, defined as follows:

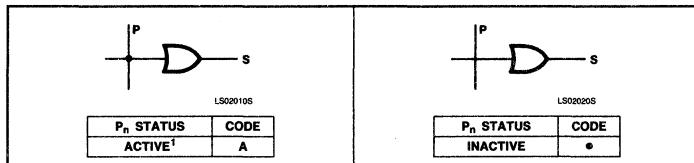
## OUTPUT POLARITY - (B)



## "AND" ARRAY - (I, B)



## OR ARRAY - (B)



### NOTES:

1. This is the initial unprogrammed state of all links.
2. Any gate P<sub>n</sub> will be unconditionally inhibited if both the true and complement of an input (either I or B) are left intact.

## VIRGIN STATE

A factory shipped virgin device contains all fusible links intact, such that:

1. All outputs at "H" polarity.
2. All P<sub>n</sub> terms are disabled.
3. All P<sub>n</sub> terms are active on all outputs.

## CAUTION: PLS153A TEST COLUMNS

The PLS153A incorporates two columns not shown in the logic block diagram. These columns are used for in-house testing of the device in the unprogrammed state. These columns must be disabled prior to using the PLS153A in your application. If you are using a Signetics-approved programmer the disabling is accomplished during the device programming sequence. If these columns are not disabled, abnormal operation is possible.

Furthermore, because of these test columns, the PLS153A cannot be programmed using the programmer algorithm for the PLS153.

4

# Field-Programmable Logic Array (18 × 42 × 10)

PLS153A

## FPLA PROGRAM TABLE

<p><b>Notes</b></p> <ol style="list-style-type: none"> <li>1. The FPLA is shipped with all links intact. Thus a background of entries corresponding to states of virgin links exists in the table. (Shown blank for clarity)</li> <li>2. Don't Care (1 and 0 bits in the AND array must be programmed Don't Care (-))</li> <li>3. Unused product terms can be left blank.</li> </ol>		<b>AND</b>		<b>OR</b> <b>B(0)</b>										<b>POLARITY</b>																												
		<b>TERM</b>		<b>B(0)</b>										<b>B(0)</b>																												
<p>CUSTOMER NAME _____</p> <p>PURCHASE ORDER # _____</p> <p>SIGNETICS DEVICE # _____ CF (XXXX)</p> <p>CUSTOMER SYMBOLIZED PART # _____</p> <p>TOTAL NUMBER OF PARTS _____</p> <p>PROGRAM TABLE # _____ REV _____ DATE _____</p>																																										
<p><b>AND</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">INACTIVE</td> <td style="text-align: center;">0</td> </tr> <tr> <td style="text-align: center;">L B</td> <td style="text-align: center;">H</td> </tr> <tr> <td style="text-align: center;">1 B</td> <td style="text-align: center;">L</td> </tr> <tr> <td style="text-align: center;">DON'T CARE</td> <td style="text-align: center;">-</td> </tr> </table>		INACTIVE	0	L B	H	1 B	L	DON'T CARE	-																																	
		INACTIVE	0																																							
		L B	H																																							
		1 B	L																																							
		DON'T CARE	-																																							
		<b>OR</b>																																								
		<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">ACTIVE</td> <td style="text-align: center;">A</td> <td style="text-align: center;">B(0)</td> </tr> <tr> <td style="text-align: center;">INACTIVE</td> <td style="text-align: center;">*</td> <td></td> </tr> </table>		ACTIVE	A	B(0)	INACTIVE	*																																		
		ACTIVE	A	B(0)																																						
		INACTIVE	*																																							
		<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">CONTROL</td> <td></td> </tr> <tr> <td style="text-align: center;">HIGH</td> <td style="text-align: center;">H</td> </tr> <tr> <td style="text-align: center;">LOW</td> <td style="text-align: center;">L</td> </tr> </table>		CONTROL		HIGH	H	LOW	L																																	
CONTROL																																										
HIGH	H																																									
LOW	L																																									
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">POL</td> <td></td> </tr> <tr> <td style="text-align: center;">HIGH</td> <td style="text-align: center;">H</td> </tr> <tr> <td style="text-align: center;">LOW</td> <td style="text-align: center;">L</td> </tr> </table>		POL		HIGH	H	LOW	L																																			
POL																																										
HIGH	H																																									
LOW	L																																									
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">D9</td> <td style="text-align: center;">D8</td> <td style="text-align: center;">D7</td> <td style="text-align: center;">D6</td> <td style="text-align: center;">D5</td> <td style="text-align: center;">D4</td> <td style="text-align: center;">D3</td> <td style="text-align: center;">D2</td> <td style="text-align: center;">D1</td> <td style="text-align: center;">D0</td> </tr> </table>		D9	D8	D7	D6	D5	D4	D3	D2	D1	D0																															
D9	D8	D7	D6	D5	D4	D3	D2	D1	D0																																	
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">PN</td> <td style="text-align: center;">8</td> <td style="text-align: center;">7</td> <td style="text-align: center;">6</td> <td style="text-align: center;">5</td> <td style="text-align: center;">4</td> <td style="text-align: center;">3</td> <td style="text-align: center;">2</td> <td style="text-align: center;">1</td> <td style="text-align: center;">10</td> <td style="text-align: center;">18</td> <td style="text-align: center;">17</td> <td style="text-align: center;">16</td> <td style="text-align: center;">15</td> <td style="text-align: center;">14</td> <td style="text-align: center;">13</td> <td style="text-align: center;">12</td> <td style="text-align: center;">11</td> <td style="text-align: center;">9</td> </tr> </table>		PN	8	7	6	5	4	3	2	1	10	18	17	16	15	14	13	12	11	9																						
PN	8	7	6	5	4	3	2	1	10	18	17	16	15	14	13	12	11	9																								
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">10</td> <td style="text-align: center;">18</td> <td style="text-align: center;">17</td> <td style="text-align: center;">16</td> <td style="text-align: center;">15</td> <td style="text-align: center;">14</td> <td style="text-align: center;">13</td> <td style="text-align: center;">12</td> <td style="text-align: center;">11</td> <td style="text-align: center;">9</td> </tr> </table>		10	18	17	16	15	14	13	12	11	9																															
10	18	17	16	15	14	13	12	11	9																																	
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">VARIABLE NAME</td> <td colspan="10"></td> <td colspan="10"></td> </tr> </table>		VARIABLE NAME																																								
VARIABLE NAME																																										

# PLHS153

## Field-Programmable Logic Array (18 × 42 × 10)

Signetics Programmable Logic  
Preliminary Specification

Application Specific Products  
• Series 20

### DESCRIPTION

The PLHS153 is a high-speed version of the PLS153 and PLS153A FPLAs. The Signetics state-of-the-art Oxide Isolated Bipolar fabrication process is employed to produce performance levels not yet achieved in devices of this complexity.

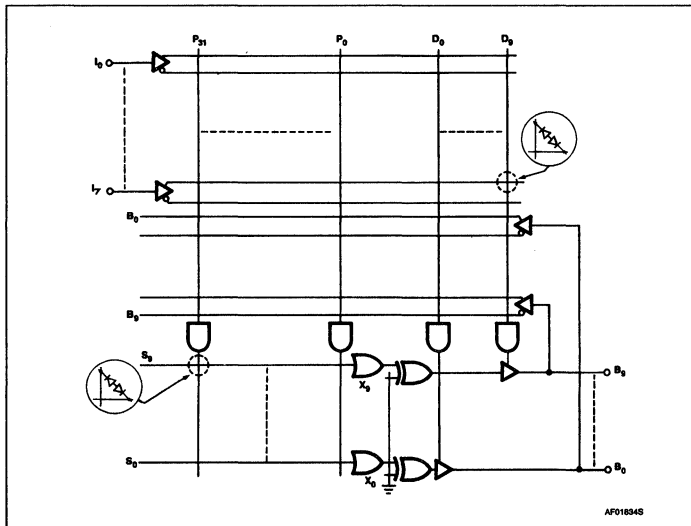
The PLHS153 is a two-level logic elements, consisting of 42 AND gates and 10 OR gates with fusible link connections for programming I/O polarity and direction.

All AND gates are linked to 8 inputs (I) and 10 bidirectional I/O lines (B). These yield variable I/O gate configurations via 10 direction control gates (D), ranging from 18 inputs to 10 outputs.

On chip T/C buffers couple either True (I, B) or Complement (I, B) input polarities to all AND gates, whose outputs can be optionally linked to all OR gates. Their output polarity, in turn, is individually programmable through a set of EX-OR gates for implementing AND/OR or AND/NOR logic functions.

The PLHS153 is field programmable, enabling the user to quickly generate custom patterns using standard programming equipment.

### FUNCTIONAL DIAGRAM



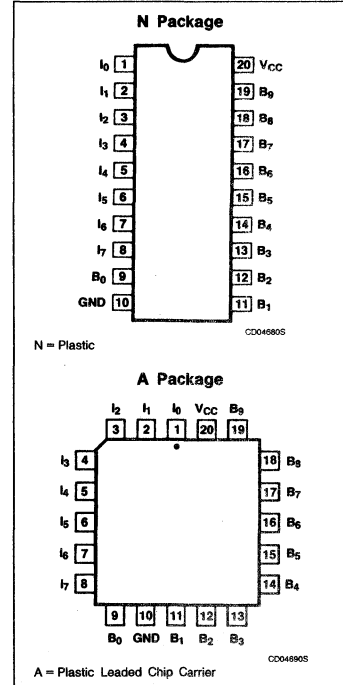
### FEATURES

- Field-Programmable
- Functionally identical to PLS153/153A
- 8 Inputs
- 42 AND gates
- 10 OR gates
- 10 bidirectional I/O lines
- Active-High or -Low outputs
- 42 product terms:
  - 32 logic terms
  - 10 control terms
- I/O propagation delay: 20ns (max.)
- Input loading: -100µA (max.)
- Power dissipation: 650mW (typ.)
- Tri-state outputs
- TTL compatible

### APPLICATIONS

- Random logic
- Code converters
- Fault detectors
- Function generators
- Address mapping
- Multiplexing

### PIN CONFIGURATIONS



### LOGIC FUNCTION

**TYPICAL PRODUCT TERM:**  
 $P_n = A \cdot B \cdot C \cdot D \dots$

**TYPICAL LOGIC FUNCTION:**  
 AT OUTPUT POLARITY = H  
 $Z = P_0 + P_1 + P_2 \dots$

AT OUTPUT POLARITY = L  
 $Z = \overline{P_0} \cdot \overline{P_1} \cdot \overline{P_2} \dots$

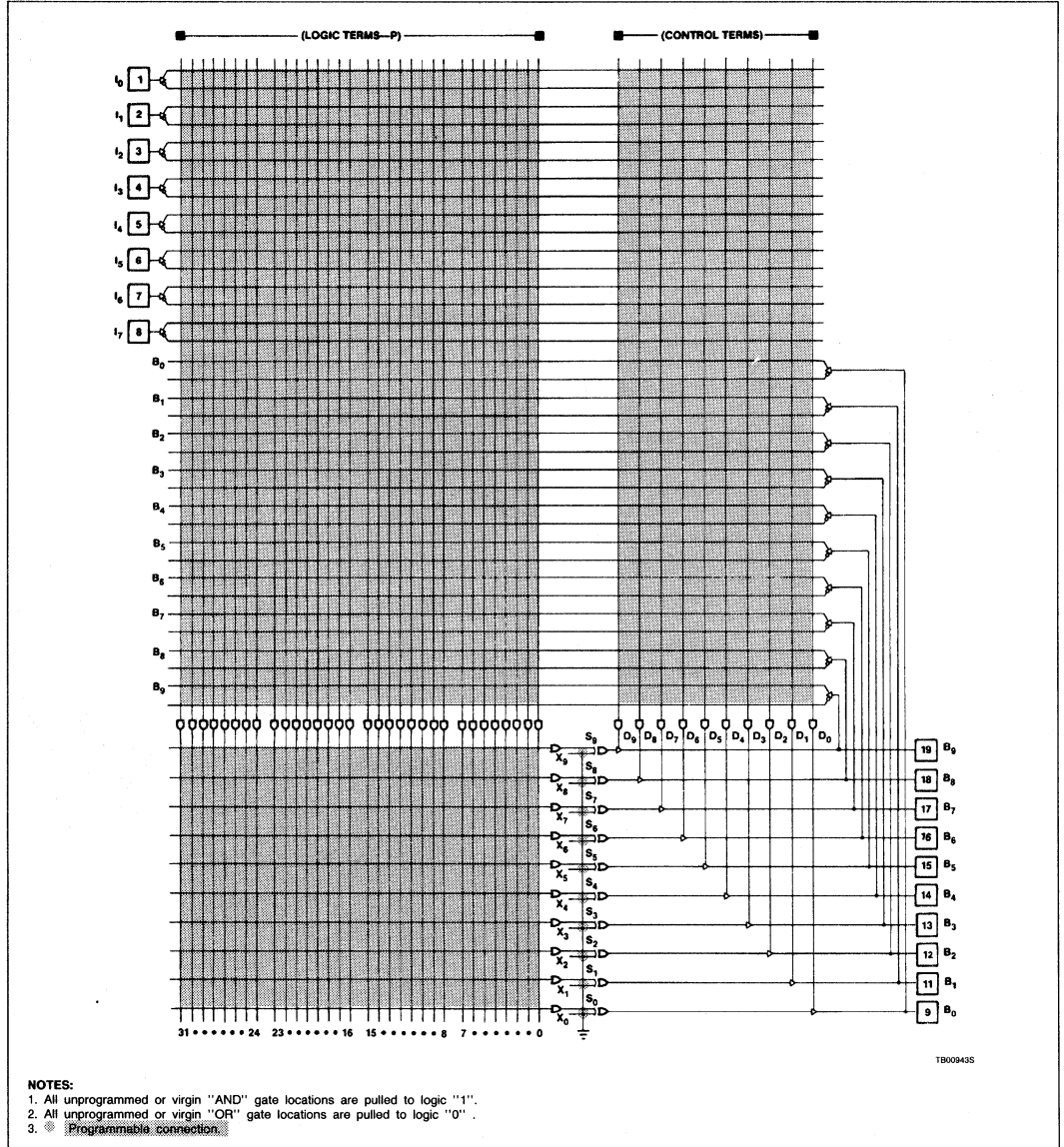
**NOTES:**

1. For each of the 10 outputs, either function Z (Active-High) or  $\overline{Z}$  (Active-Low) is available, but not both. The desired output polarity is programmed via the EX-OR gates.
2. Z, A, B, C, etc. are user defined connections to fixed inputs (I) and bidirectional pins (B).

# Field-Programmable Logic Array (18 × 42 × 10)

PLHS153

## FPLA LOGIC DIAGRAM



## Field-Programmable Logic Array (18 × 42 × 10)

PLHS153

## ORDERING INFORMATION

DESCRIPTION	ORDER CODE
20-pin Plastic DIP 300mil-wide	PLHS153N
20-pin Plastic Leaded Chip Carrier	PLHS153A

ABSOLUTE MAXIMUM RATINGS<sup>1</sup>

SYMBOL	PARAMETER	RATING		UNIT
		Min	Max	
V <sub>CC</sub>	Supply voltage		+7	V <sub>DC</sub>
V <sub>IN</sub>	Input voltage		+5.5	V <sub>DC</sub>
V <sub>OUT</sub>	Output voltage		+5.5	V <sub>DC</sub>
I <sub>IN</sub>	Input currents	-30	+30	mA
I <sub>OUT</sub>	Output currents		+100	mA
T <sub>A</sub> T <sub>STG</sub>	Temperature range Operating Storage	0 -65	+75 +150	°C

## THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

The PLHS153 device is also processed to military requirements for operation over the military temperature range. For specifications and ordering information consult the Signetics Military Data Book.

4

DC ELECTRICAL CHARACTERISTICS 0°C ≤ T<sub>A</sub> ≤ 75°C, 4.75V ≤ V<sub>CC</sub> ≤ 5.25V

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ <sup>2</sup>	Max	
<b>Input voltage<sup>3</sup></b>						
V <sub>IL</sub> V <sub>IH</sub> V <sub>IC</sub>	Low High Clamp <sup>3,4</sup>	V <sub>CC</sub> = Min V <sub>CC</sub> = Max V <sub>CC</sub> = Min, I <sub>IN</sub> = -12mA	2.0	-0.8	0.8 -1.2	V
<b>Output voltage</b>						
V <sub>OL</sub> V <sub>OH</sub>	Low <sup>3,5</sup> High <sup>3,6</sup>	V <sub>CC</sub> = Min I <sub>OL</sub> = 15mA I <sub>OH</sub> = -2mA	2.4		0.5	V
<b>Input current<sup>11</sup></b>						
I <sub>IL</sub> I <sub>IH</sub>	Low High	V <sub>CC</sub> = Max V <sub>IN</sub> = 0.45V V <sub>IN</sub> = 5.5V			-100 40	μA
<b>Output current</b>						
I <sub>O(OFF)</sub> I <sub>CS</sub>	Hi-Z state <sup>10</sup> Short circuit <sup>4,6,7</sup>	V <sub>CC</sub> = Max V <sub>OUT</sub> = 5.5V V <sub>OUT</sub> = 0.45V V <sub>OUT</sub> = 0V	-15		80 -140 -70	μA mA
I <sub>CC</sub>	V <sub>CC</sub> supply current <sup>8</sup>	V <sub>CC</sub> = Max		130	155	mA
<b>Capacitance</b>						
C <sub>IN</sub> C <sub>B</sub>	Input I/O	V <sub>CC</sub> = 5V V <sub>IN</sub> = 2.0V V <sub>B</sub> = 2.0V		8 15		pF

Notes on following page.



# Field-Programmable Logic Array (18 × 42 × 10)

PLHS153

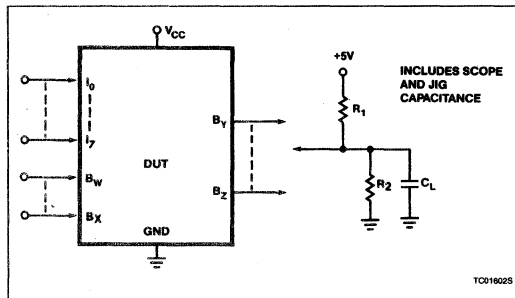
## AC ELECTRICAL CHARACTERISTICS $R_1 = 470\Omega$ , $R_2 = 1k\Omega$ , $0^\circ C \leq T_A \leq +75^\circ C$ , $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TO	FROM	TEST CONDITIONS	LIMITS			UNIT
					Min	Typ <sup>2</sup>	Max	
$t_{PD}$	Propagation delay	Output±	Input±	$C_L = 30pF$		14	20	ns
$t_{OE}$	Output enable	Output-	Input±	$C_L = 30pF$		16	20	ns
$t_{OD}$	Output disable <sup>9</sup>	Output+	Input±	$C_L = 5pF$		14	20	ns

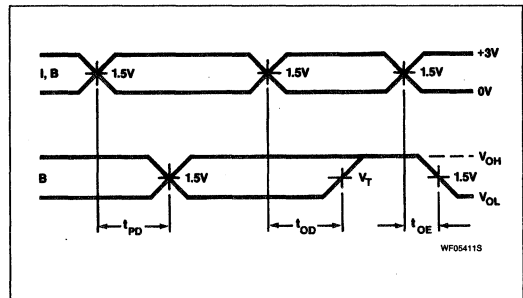
**NOTES:**

1. Stresses above those listed under "Absolute Maximum Ratings" may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other conditions above those indicated in the operational and programming specification of the device is not implied.
2. All typical values are at  $V_{CC} = 5V$ ,  $T_A = +25^\circ C$ .
3. All voltage values are with respect to network ground terminal.
4. Test one at a time.
5. Measured with +10V applied to  $I_S$ .
6. Output sink current is supplied through a resistor to  $V_{CC}$ .
7. Duration of short circuit should not exceed 1 second.
8.  $I_{CC}$  is measured with all pins = open or 4.5V.
9. Measured at  $V_T = V_{OL} + 0.5V$ .
10. Leakage values are a combination of input and output leakage.
11.  $I_{IL}$  and  $I_{IH}$  limits are for dedicated inputs only ( $I_0-I_7$ ).

### TEST LOAD CIRCUIT



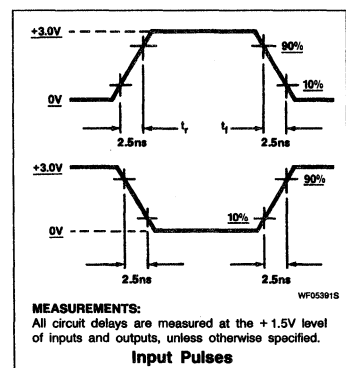
### TIMING DIAGRAM



### TIMING DEFINITIONS

- $t_{PD}$  Propagation delay between input and output.
- $t_{OD}$  Delay between input change and when output is off (Hi-Z or High).
- $t_{OE}$  Delay between input change and when output reflects specified output level.

### VOLTAGE WAVEFORM



# Field-Programmable Logic Array (18 × 42 × 10)

PLHS153

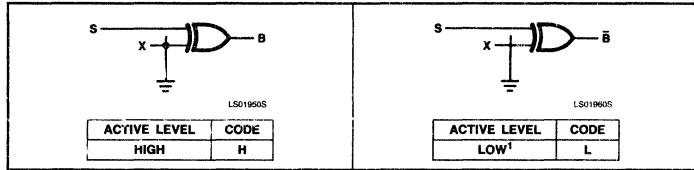
## LOGIC PROGRAMMING

The FPLA can be programmed by means of Logic programming equipment.

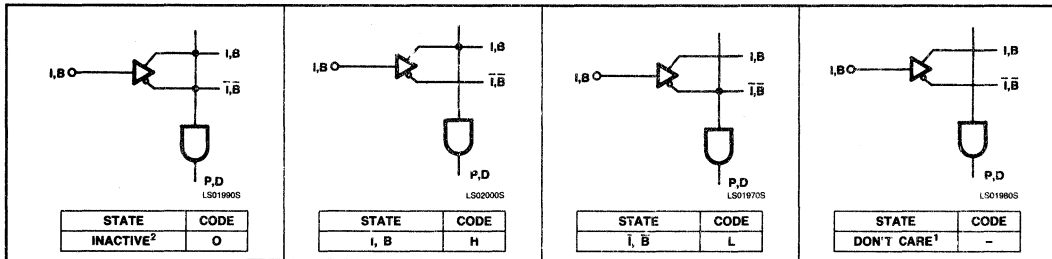
With Logic programming, the AND/OR/EX-OR gate input connections necessary to implement the desired logic function are coded directly from logic equations using the Program Table on the following page.

In this Table the logic state of variables I, P, and B associated with each Sum Term S is assigned a symbol which results in the proper fusing pattern of corresponding links defined as follows:

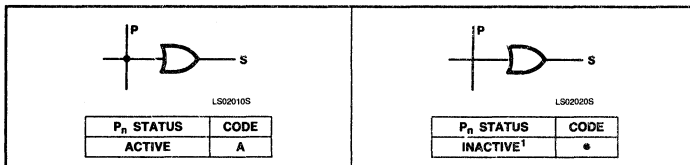
## OUTPUT POLARITY - (B)



## "AND" ARRAY - (I, B)



## OR ARRAY - (B)



### NOTES:

1. This is the initial unprogrammed state of all links. It is normally associated with all unused (inactive) AND gates P<sub>n</sub>, D<sub>n</sub>.
2. Any gate P<sub>n</sub> will be unconditionally inhibited if both the true and complement of an input (either I or B) are programmed for a connection.

## VIRGIN STATE

A factory shipped virgin device contains all fusible links intact, such that:

1. All outputs at "L" polarity.
2. All P<sub>n</sub> terms are enabled.
3. All P<sub>n</sub> terms are inactive on all outputs.

4

# Field-Programmable Logic Array (18 × 42 × 10)

PLHS153

## FPLA PROGRAM TABLE

AND		AND														OR													
		B(i)														B(0)													
		7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0
0																													
1																													
2																													
3																													
4																													
5																													
6																													
7																													
8																													
9																													
10																													
11																													
12																													
13																													
14																													
15																													
16																													
17																													
18																													
19																													
20																													
21																													
22																													
23																													
24																													
25																													
26																													
27																													
28																													
29																													
30																													
31																													
D8																													
D8																													
D7																													
D6																													
D5																													
D4																													
D3																													
D2																													
D1																													
D0																													
PIN		8	7	6	5	4	3	2	1	19	18	17	16	15	14	13	12	11	9	19	18	17	16	15	14	13	12	11	9
VARIABLE NAME																													

**Notes**

- The FPLA is shipped with all lines open.
- Inputs in the AND array exist as Don't Care (X) in the OR array.
- All P-terms are inactive on all outputs (B(i) in the high array.
- Unused product terms can be left blank.

**Customer Information**

CUSTOMER NAME \_\_\_\_\_

PURCHASE ORDER # \_\_\_\_\_

SIGNETICS DEVICE # \_\_\_\_\_ CF (XXXX)

CUSTOMER SYMBOLIZED PART # \_\_\_\_\_

TOTAL NUMBER OF PARTS \_\_\_\_\_

PROGRAM TABLE # \_\_\_\_\_ REV \_\_\_\_\_ DATE \_\_\_\_\_

**AND/OR Controls**

OR: ACTIVE (A) B(0) INACTIVE (X)

AND: ACTIVE (A) B(0) INACTIVE (X)

CONTROL: HIGH (H) LOW (L) (POL)

**AND/OR Controls**

AND: INACTIVE (0) H L DON'T CARE (-)

OR: INACTIVE (0) H L DON'T CARE (-)

TB00955

## PLS155 Field-Programmable Logic Sequencer (16 × 45 × 12)

Signetics Programmable Logic  
Product Specification

### Application Specific Products • Series 20

#### DESCRIPTION

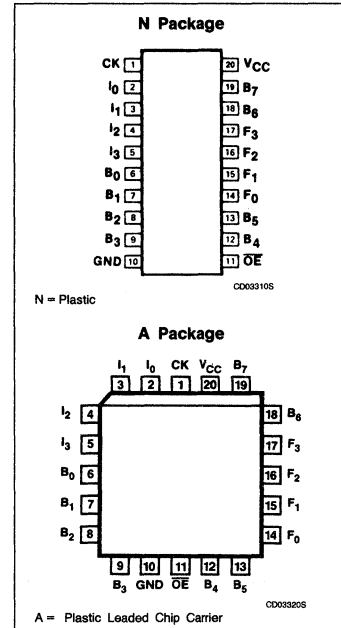
The PLS155 is a Tri-state output, registered logic element combining AND/OR gate arrays with clocked J-K flip-flops. These J-K flip-flops are dynamically convertible to D-type via a "fold-back" inverting buffer and control gate  $F_C$ . It features 4 registered I/O outputs (F) in conjunction with 8 bidirectional I/O lines (B). These yield variable I/O gate and register configurations via control gates (D, L) ranging from 16 inputs to 12 outputs.

The AND/OR arrays consist of 32 logic AND gates, 13 control AND gates, and 21 OR gates with fusible link connections for programming I/O polarity and direction. All AND gates are linked to 4 inputs (I), bidirectional I/O lines (B), internal flip-flop outputs (Q), and Complement Array output ( $\bar{C}$ ). The Complement Array consists of a NOR gate optionally linked to all AND gates for generating and propagating complementary AND terms.

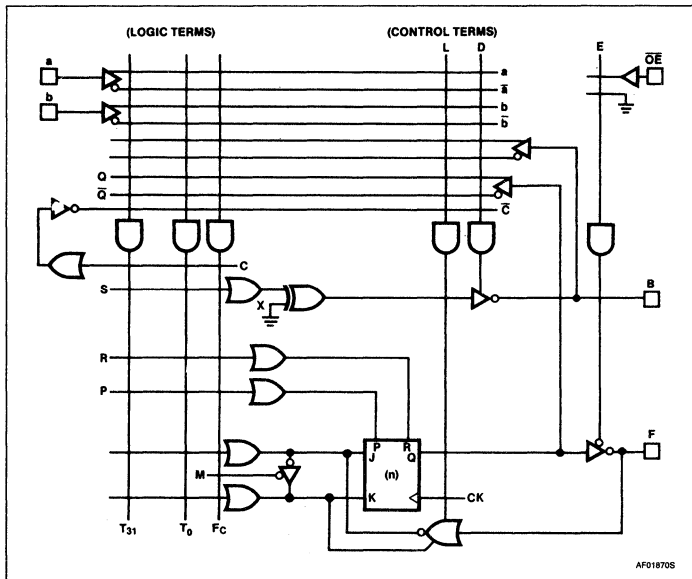
#### FEATURES

- Field-Programmable (Ni-Cr link)
- 4 dedicated inputs
- 13 control gates
- 32 AND gates
- 21 OR gates
- 45 product terms:
  - 32 logic terms
  - 13 control terms
- 8 bidirectional I/O lines
- 4 bidirectional registers
- J-K, T, or D-type flip-flops
- Asynchronous Preset/Reset
- Complement Array
- Active-High or -Low outputs.
- Programmable  $\bar{O}E$  control
- Positive edge-triggered clock
- Clock frequency: 15MHz (max.)
- Input loading:  $-100\mu A$  (max.)
- Power dissipation: 750mW (typ.)
- TTL compatible
- Tri-state outputs

#### PIN CONFIGURATIONS



#### FUNCTIONAL DIAGRAM



#### APPLICATIONS

- Random sequential logic
- Synchronous up/down counters
- Shift registers
- Bidirectional data buffers
- Timing function generators
- System controllers/synchronizers
- Priority encoder/registers

# Field-Programmable Logic Sequencer (16 × 45 × 12)

PLS155

On-chip T/C buffers couple either True (I, B, Q) or Complement (I, B, Q, C) input polarities to all AND gates, whose outputs can be optionally linked to all OR gates. One group of AND gates drives bidirectional I/O lines (B), whose output polarity is individually programmable through a set of EX-OR gates for implementing AND-OR or AND-NOR logic functions. Another group drives the J-K inputs of all flip-flops. The Asynchronous Preset and Reset lines (P, R), are driven from the OR matrix.

All flip-flops are positive edge-triggered and can be used as input, output or I/O (for interfacing with a bidirectional data bus) in conjunction with load control gates (L), steering inputs (I), (B), (Q) and programmable output select lines (E).

The PLS155 is field programmable, enabling the user to quickly generate custom patterns using standard programming equipment.

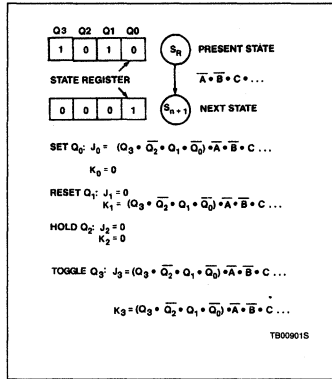
Order codes are contained on the pages following.

### VIRGIN STATE

The factory shipped virgin device contains all fusible links intact, such that:

1.  $\overline{OE}$  is always enabled.
2. Preset and Reset are always disabled.
3. All transition terms are disabled.
4. All flip-flops are in D-mode unless otherwise programmed to J/K only or J/K or D (controlled).
5. All B pins, are inputs and all F pins are outputs unless otherwise programmed.

### LOGIC FUNCTION



**NOTES:**  
 Similar logic functions are applicable for D and T mode flip-flops.

### FLIP-FLOP TRUTH TABLE

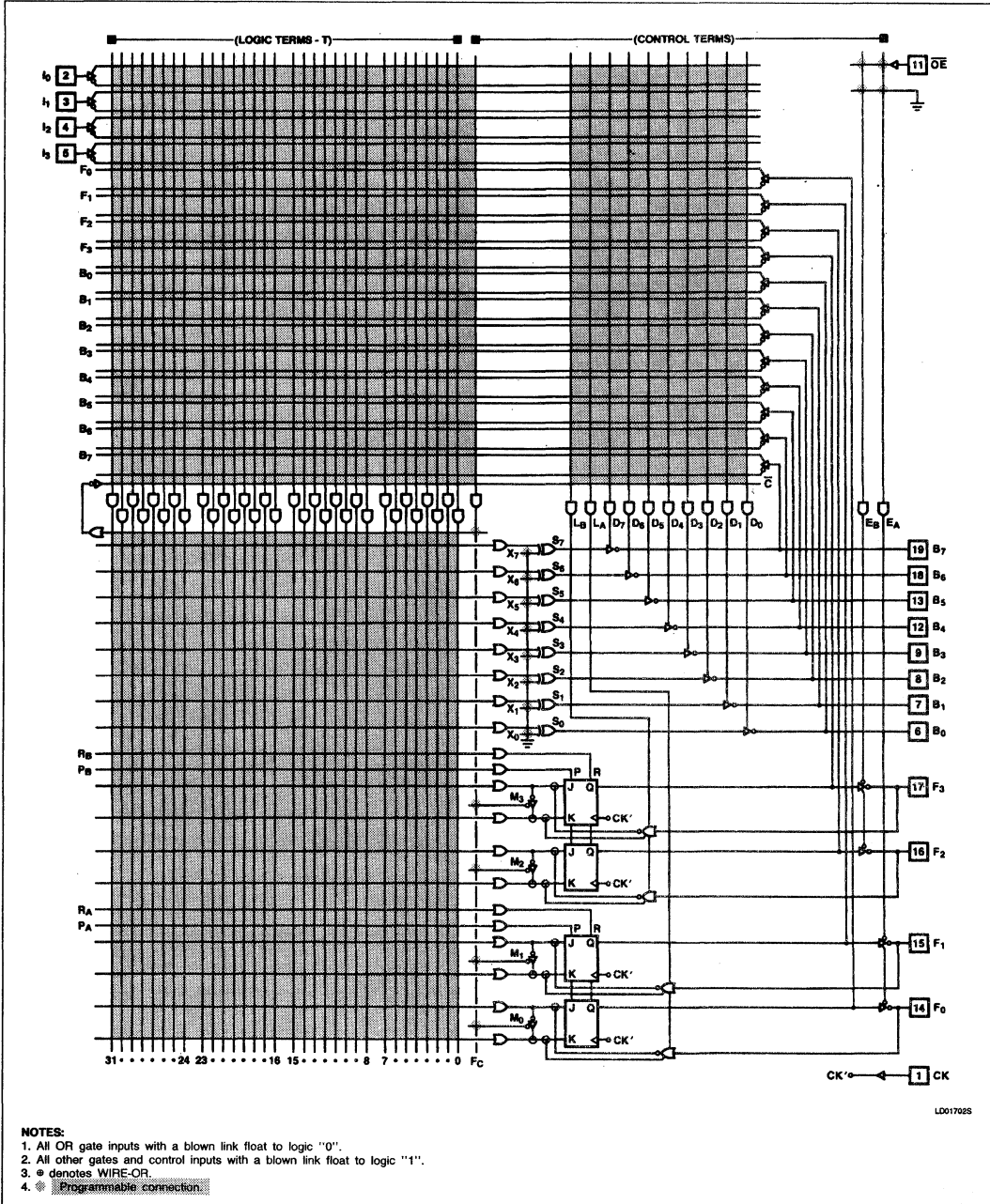
$\overline{OE}$	L	C	P	R	J	K	Q	F
H	X							HI-Z
L	X	X	H	L	X	X	H	L
L	X	X	L	H	X	X	L	H
L	L	↑	L	L	L	L	Q	$\overline{Q}$
L	L	↑	L	L	L	H	L	H
L	L	↑	L	L	H	L	H	L
L	L	↑	L	L	H	H	$\overline{Q}$	Q
H	H	↑	L	L	L	H	L	H*
H	H	↑	L	L	H	L	H	L*
+10V	X	↑	X	X	L	H	L	H**
	X	↑	X	X	H	L	H	L**

- NOTES:**
1. Positive Logic:  
 $J/K = T_0 + T_1 + T_2 \dots T_{31}$   
 $T_n = \overline{C} \cdot (I_0 \cdot I_1 \cdot I_2 \dots) \cdot (Q_0 \cdot Q_1 \dots) \cdot (B_0 \cdot B_1 \dots)$
  2. I denotes transition from Low to High level.
  3. X = Don't care
  4. \* = Forced at  $F_n$  pin for loading J/K flip-flop in I/O mode. L must be enabled, and other active  $T_n$  disabled via steering input(s) I, B, or Q.
  5. At  $P = R = H$ ,  $Q = H$ . The final state of Q depends on which is released first.
  6. \*\* = Forced at  $F_n$  pin to load J/K flip-flop independent of program code (Diagnostic mode), Tri-state B outputs.

# Field-Programmable Logic Sequencer (16 × 45 × 12)

PLS155

## FPLS LOGIC DIAGRAM



4

- NOTES:**
1. All OR gate inputs with a blown link float to logic "0".
  2. All other gates and control inputs with a blown link float to logic "1".
  3. ● denotes WIRE-OR.
  4. ⊕ Programmable connection.

LD017025

## Field-Programmable Logic Sequencer (16 × 45 × 12)

PLS155

## ORDERING INFORMATION

DESCRIPTION	ORDER CODE
20-pin Plastic DIP 300mil-wide	PLS155N
20-pin Plastic Leaded Chip Carrier	PLS155A

## THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS<sup>1</sup>

SYMBOL	PARAMETER	RATING		UNIT
		Min	Max	
V <sub>CC</sub>	Supply voltage		+7	V <sub>DC</sub>
V <sub>IN</sub>	Input voltage		+5.5	V <sub>DC</sub>
V <sub>OUT</sub>	Output voltage		+5.5	V <sub>DC</sub>
I <sub>IN</sub>	Input currents	-30	+30	mA
I <sub>OUT</sub>	Output currents		+100	mA
T <sub>A</sub> T <sub>STG</sub>	Temperature range Operating Storage	0 -65	+75 +150	°C

DC ELECTRICAL CHARACTERISTICS 0°C ≤ T<sub>A</sub> ≤ +75°C, 4.75 ≤ V<sub>CC</sub> ≤ 5.25V

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ <sup>2</sup>	Max	
<b>Input voltage<sup>3</sup></b>						
V <sub>IH</sub> V <sub>IL</sub> V <sub>IC</sub>	High Low Clamp	V <sub>CC</sub> = Max V <sub>CC</sub> = Min V <sub>CC</sub> = Min, I <sub>IN</sub> = -12mA	2		0.8 -1.2	V
<b>Output voltage<sup>3</sup></b>						
V <sub>OH</sub> V <sub>OL</sub>	High Low	V <sub>CC</sub> = Min I <sub>OH</sub> = -2mA I <sub>OL</sub> = 10mA	2.4	0.35	0.5	V
<b>Input current<sup>6</sup></b>						
I <sub>IH</sub> I <sub>IL</sub>	High Low	V <sub>CC</sub> = Max V <sub>IN</sub> = 5.5V V <sub>IN</sub> = 0.45V		< 1 -10	80 -100	μA
<b>Output current</b>						
I <sub>O</sub> (OFF) I <sub>OS</sub>	Hi-Z state <sup>8</sup> Short circuit <sup>4</sup>	V <sub>CC</sub> = Max V <sub>OUT</sub> = 5.5V V <sub>OUT</sub> = 0.45V V <sub>OUT</sub> = 0V		1 -1	80 -140 -70	μA mA
I <sub>CC</sub>	V <sub>CC</sub> supply current <sup>7</sup>	V <sub>CC</sub> = Max		150	190	mA
<b>Capacitance</b>						
C <sub>IN</sub> C <sub>OUT</sub>	Input Output	V <sub>CC</sub> = 5.0V V <sub>IN</sub> = 2.0V V <sub>OUT</sub> = 2.0V		8 15		pF

## NOTES:

- Stresses above those listed under "Absolute Maximum Ratings" may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.
- All typical values are at V<sub>CC</sub> = 5V, T<sub>A</sub> = +25°C.
- All voltage values are with respect to network ground terminal.
- Test one at a time.
- Measured with V<sub>IH</sub> applied to  $\overline{OE}$ .
- Duration of short circuit should not exceed 1 second.
- I<sub>CC</sub> is measured with the  $\overline{OE}$  input grounded, all other inputs at 4.5V, and the outputs open.
- Leakage values are a combination of input and output leakage.

## Field-Programmable Logic Sequencer (16 × 45 × 12)

PLS155

**AC ELECTRICAL CHARACTERISTICS** 0°C ≤ T<sub>A</sub> ≤ +75°C, 4.75 ≤ V<sub>CC</sub> ≤ 5.25V, R<sub>1</sub> = 470Ω, R<sub>2</sub> = 1kΩ

SYMBOL	PARAMETER	TO	FROM	TEST CONDITIONS	LIMITS			UNIT
					Min <sup>5</sup>	Typ <sup>1</sup>	Max	
<b>Pulse width</b>								
t <sub>CKH</sub>	Clock <sup>2</sup> high	CK -	CK +	C <sub>L</sub> = 30pF	25	20		ns
t <sub>CKL</sub>	Clock low	CK +	CK -		30	20		
t <sub>CKP</sub>	Period	CK +	CK +		70	50		
t <sub>PRH</sub>	Preset/Reset pulse	(I,B) +	(I,B) -		40	30		
<b>Setup time</b>								
t <sub>IS1</sub>	Input	CK +	(I,B) ±	C <sub>L</sub> = 30pF	40	30		ns
t <sub>IS2</sub>	Input (through F <sub>n</sub> )	CK +	F ±		20	10		
t <sub>IS3</sub>	Input (through Complement Array) <sup>4</sup>	CK +	(I,B) ±		65	40		
<b>Hold time</b>								
t <sub>IH1</sub>	Input	CK +	(I,B) ±	C <sub>L</sub> = 30pF	0	-10		ns
t <sub>IH2</sub>		CK +	F ±		15	10		
<b>Propagation delays</b>								
t <sub>CKO</sub>	Clock	F ±	CK +	C <sub>L</sub> = 30pF		25	30	ns
t <sub>OE1</sub>	Output enable	F -	$\overline{OE}$ -	C <sub>L</sub> = 30pF		20	30	
t <sub>OD1</sub>	Output disable <sup>3</sup>	F +	$\overline{OE}$ +	C <sub>L</sub> = 5pF		20	30	
t <sub>PD</sub>	Output	B ±	(I,B) ±	C <sub>L</sub> = 30pF		40	50	
t <sub>OE2</sub>	Output enable	B ±	(I,B) +	C <sub>L</sub> = 30pF		35	55	
t <sub>OD2</sub>	Output disable <sup>3</sup>	B +	(I,B) -	C <sub>L</sub> = 5pF		30	35	
t <sub>PRO</sub>	Preset/Reset	F ±	(I,B) +	C <sub>L</sub> = 30pF		50	55	

**NOTES:**

- All typical values are at V<sub>CC</sub> = 5V, T<sub>A</sub> = +25°C.
- To prevent spurious clocking, clock rise time (10% - 90%) ≤ 10ns.
- Measured at V<sub>T</sub> = V<sub>OL</sub> + 0.5V.
- When using the Complement Array T<sub>CKP</sub> = 95ns (min).
- Limits are guaranteed with 12 product terms maximum connected to each sum term line.
- For test circuits, waveforms and timing diagrams see the following pages.

---



---

# 4

---



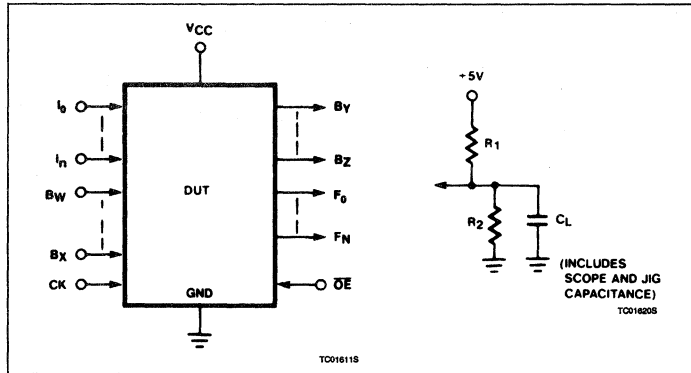
---



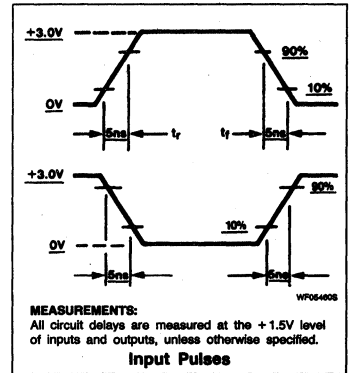
# Field-Programmable Logic Sequencer (16 × 45 × 12)

PLS155

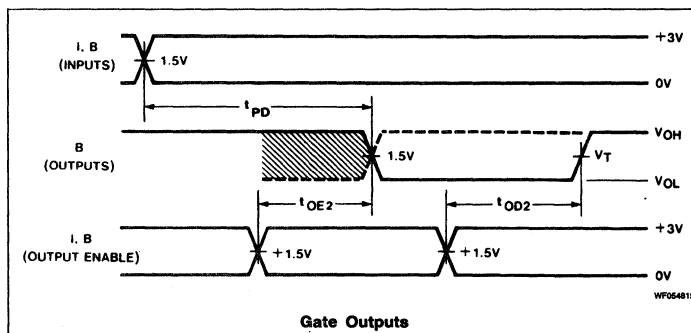
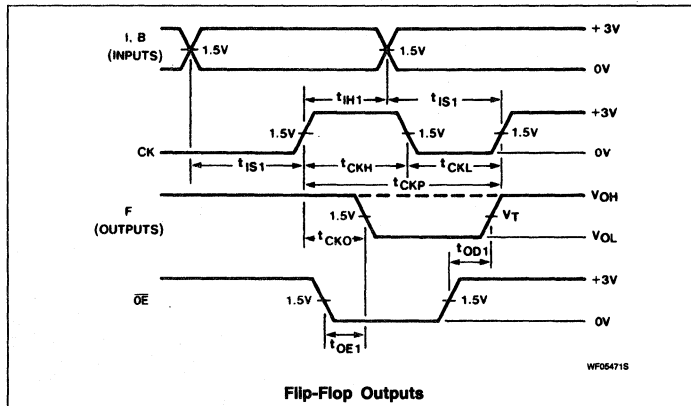
## TEST LOAD CIRCUIT



## VOLTAGE WAVEFORM



## TIMING DIAGRAMS



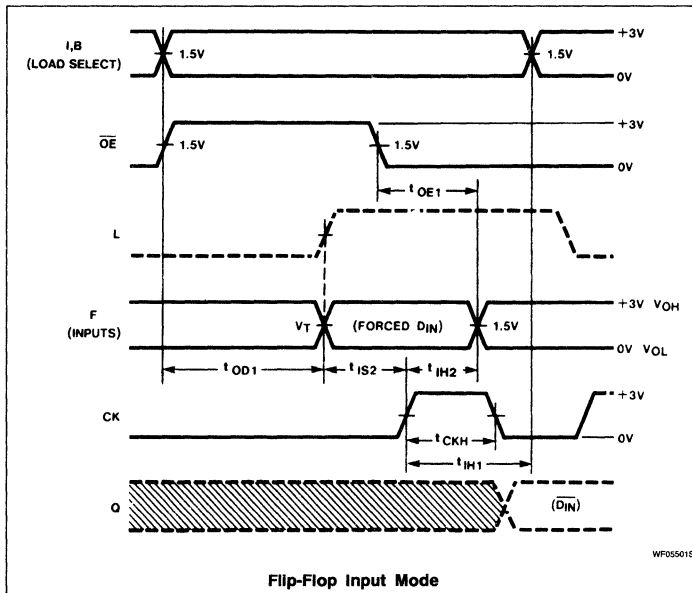
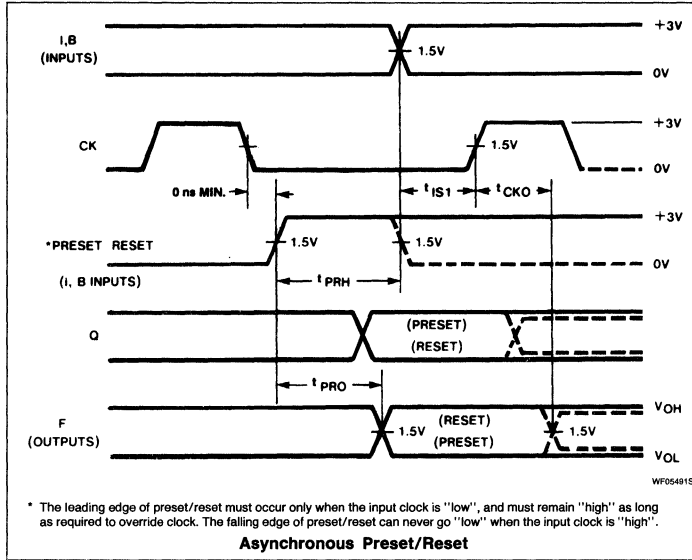
## TIMING DEFINITIONS

- $t_{CKH}$  Width of input clock pulse.
- $t_{CKL}$  Interval between clock pulses.
- $t_{CKP}$  Clock period.
- $t_{PRH}$  Width of preset input pulse.
- $t_{IS1}$  Required delay between beginning of valid input and positive transition of clock.
- $t_{IS2}$  Required delay between beginning of valid input forced at flip-flop output pins, and positive transition of clock.
- $t_{IH1}$  Required delay between positive transition of clock and end of valid input data.
- $t_{IH2}$  Required delay between positive transition of clock and end of valid input data forced at flip-flop output pins.
- $t_{CKO}$  Delay between positive transition of clock and when Outputs become valid (with OE low).
- $t_{OE1}$  Delay between beginning of Output Enable Low and when Outputs become valid.
- $t_{OD1}$  Delay between beginning of Output Enable High and when Outputs are in the off state.
- $t_{PD}$  Propagation delay between combinational inputs and outputs.
- $t_{OE2}$  Delay between predefined Output Enable High, and when combinational Outputs become valid.
- $t_{OD2}$  Delay between predefined Output Enable Low, and when combinational Outputs are in the off state.
- $t_{PRO}$  Delay between positive transition of predefined Preset/Reset input, and when flip-flop outputs become valid.

Field-Programmable Logic Sequencer (16 × 45 × 12)

PLS155

TIMING DIAGRAMS (Continued)



4

# Field-Programmable Logic Sequencer (16 × 45 × 12)

PLS155

The FPLS can be programmed by means of Logic Programming equipment.

With Logic programming, the AND/OR-EX-OR input connections necessary to imple-

ment the desired logic function are coded directly from the State Diagram using the Program Tables on the following pages.

In these tables, the logic state or action of all I/O, control, and state variables is assigned a symbol which results in the proper fusing pattern of corresponding links, defined as follows:

## "AND" ARRAY - (I), (B), (Q<sub>p</sub>)

<p>(T, F<sub>C</sub>, L, P, R, D)<sub>n</sub></p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><th>STATE</th><th>CODE</th></tr> <tr><td>INACTIVE<sup>1,2</sup></td><td>O</td></tr> </table> <p style="font-size: small; text-align: center;">LS020305</p>	STATE	CODE	INACTIVE <sup>1,2</sup>	O	<p>(T, F<sub>C</sub>, L, P, R, D)<sub>n</sub></p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><th>STATE</th><th>CODE</th></tr> <tr><td>I, B, Q</td><td>H</td></tr> </table> <p style="font-size: small; text-align: center;">LS020405</p>	STATE	CODE	I, B, Q	H	<p>(T, F<sub>C</sub>, L, P, R, D)<sub>n</sub></p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><th>STATE</th><th>CODE</th></tr> <tr><td>I, B, Q</td><td>L</td></tr> </table> <p style="font-size: small; text-align: center;">LS020505</p>	STATE	CODE	I, B, Q	L	<p>(T, F<sub>C</sub>, L, P, R, D)<sub>n</sub></p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><th>STATE</th><th>CODE</th></tr> <tr><td>DON'T CARE</td><td>-</td></tr> </table> <p style="font-size: small; text-align: center;">LS020605</p>	STATE	CODE	DON'T CARE	-
STATE	CODE																		
INACTIVE <sup>1,2</sup>	O																		
STATE	CODE																		
I, B, Q	H																		
STATE	CODE																		
I, B, Q	L																		
STATE	CODE																		
DON'T CARE	-																		

## "COMPLEMENT" ARRAY - (C)

<p>(T<sub>n</sub>, F<sub>C</sub>)</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><th>ACTION</th><th>CODE</th></tr> <tr><td>INACTIVE<sup>1,3,5</sup></td><td>O</td></tr> </table> <p style="font-size: small; text-align: center;">LS020705</p>	ACTION	CODE	INACTIVE <sup>1,3,5</sup>	O	<p>(T<sub>n</sub>, F<sub>C</sub>)</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><th>ACTION</th><th>CODE</th></tr> <tr><td>GENERATE<sup>5</sup></td><td>A</td></tr> </table> <p style="font-size: small; text-align: center;">LS020805</p>	ACTION	CODE	GENERATE <sup>5</sup>	A	<p>(T<sub>n</sub>, F<sub>C</sub>)</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><th>ACTION</th><th>CODE</th></tr> <tr><td>PROPAGATE</td><td>•</td></tr> </table> <p style="font-size: small; text-align: center;">LS020905</p>	ACTION	CODE	PROPAGATE	•	<p>(T<sub>n</sub>, F<sub>C</sub>)</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><th>ACTION</th><th>CODE</th></tr> <tr><td>TRANSPARENT</td><td>-</td></tr> </table> <p style="font-size: small; text-align: center;">LS021005</p>	ACTION	CODE	TRANSPARENT	-
ACTION	CODE																		
INACTIVE <sup>1,3,5</sup>	O																		
ACTION	CODE																		
GENERATE <sup>5</sup>	A																		
ACTION	CODE																		
PROPAGATE	•																		
ACTION	CODE																		
TRANSPARENT	-																		

## "OR" ARRAY - (MODE)

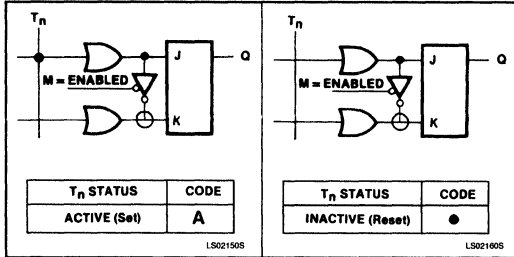
<p>(D<sub>n</sub>, L<sub>n</sub>, P<sub>n</sub>, R<sub>n</sub>)</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><th>ACTION</th><th>CODE</th></tr> <tr><td>PROPAGATE</td><td>•</td></tr> </table> <p style="font-size: small; text-align: center;">LS021105</p>	ACTION	CODE	PROPAGATE	•	<p>(D<sub>n</sub>, L<sub>n</sub>, P<sub>n</sub>, R<sub>n</sub>)</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><th>ACTION</th><th>CODE</th></tr> <tr><td>TRANSPARENT</td><td>-</td></tr> </table> <p style="font-size: small; text-align: center;">LS021205</p>	ACTION	CODE	TRANSPARENT	-
ACTION	CODE								
PROPAGATE	•								
ACTION	CODE								
TRANSPARENT	-								
<p>J K OR D (CONTROLLED)</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><th>ACTION</th><th>CODE</th></tr> <tr><td>J-K OR D (CONTROLLED)</td><td>A</td></tr> </table> <p style="font-size: small; text-align: center;">LS021305</p>	ACTION	CODE	J-K OR D (CONTROLLED)	A	<p>J-K</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><th>ACTION</th><th>CODE</th></tr> <tr><td>J-K</td><td>•</td></tr> </table> <p style="font-size: small; text-align: center;">LS021405</p>	ACTION	CODE	J-K	•
ACTION	CODE								
J-K OR D (CONTROLLED)	A								
ACTION	CODE								
J-K	•								

Notes on following page.

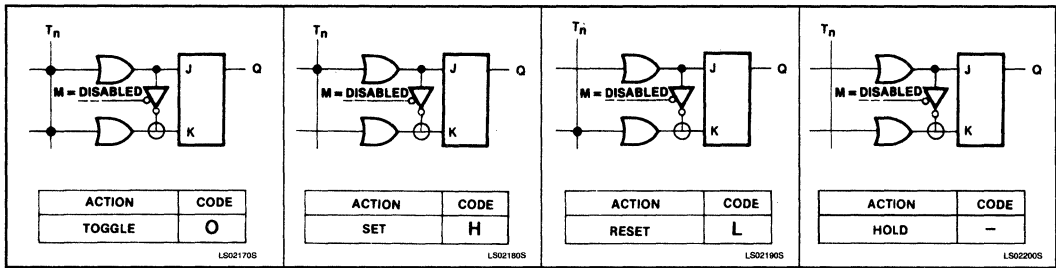
# Field-Programmable Logic Sequencer (16 × 45 × 12)

PLS155

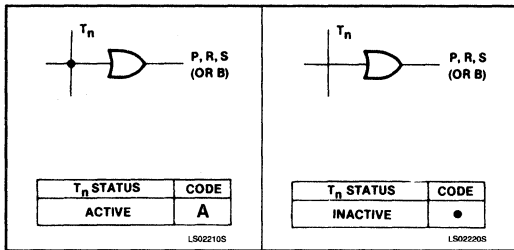
## "OR" ARRAY - ( $Q_n = D$ -Type)



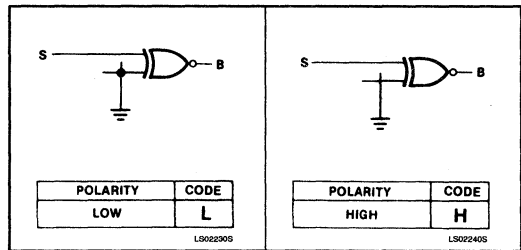
## "OR" ARRAY - ( $Q_n = J$ -K Type)



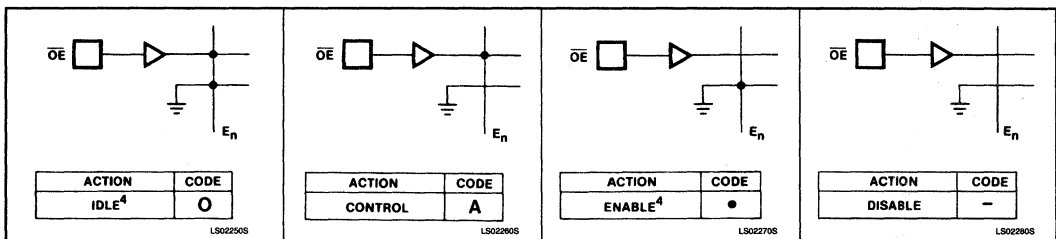
## "OR" ARRAY - (S or B), (P), (R)



## "EX-OR" ARRAY - (B)



## "OE" ARRAY - (E)



**NOTES:**

1. This is the initial unprogrammed state of all link pairs. It is normally associated with all unused (inactive) AND gates.
2. Any gate ( $T_n, F_C, L, P, R, D_n$ ) will be unconditionally inhibited if both of the I, B, or Q links are left intact.
3. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates  $T_n, F_C$ .
4.  $E_n = O$  and  $E_n = \bullet$  are logically equivalent states, since both cause  $F_n$  outputs to be unconditionally enabled.
5. These states are not allowed for control gates ( $L, P, R, D_n$ ) due to their lack of "OR" array links.

4



## PLS157 Field-Programmable Logic Sequencer (16 × 45 × 12)

Signetics Programmable Logic  
Product Specification

### Application Specific Products • Series 20

#### DESCRIPTION

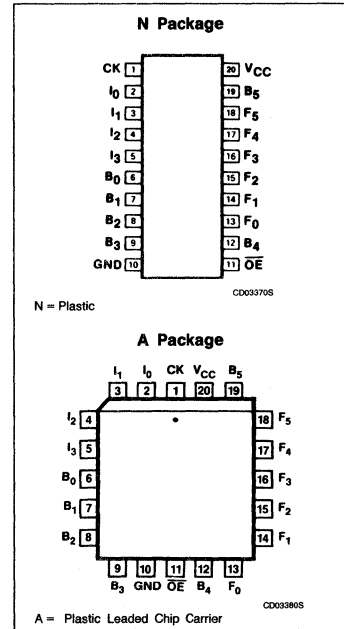
The PLS157 is a Tri-state output, registered logic element combining AND/OR gate arrays with clocked J-K flip-flops. These J-K flip-flops are dynamically convertible to D-type via a "fold-back" inverting buffer and control gate  $F_C$ . It features 6 registered I/O outputs (F) in conjunction with 6 bidirectional I/O lines (B). These yield variable I/O gate and register configurations via control gates (D, L) ranging from 16 inputs to 12 outputs.

The AND/OR arrays consist of 32 logic AND gates, 13 control AND gates, and 21 OR gates with fusible link connections for programming I/O polarity and direction. All AND gates are linked to 4 inputs (I), bidirectional I/O lines (B), internal flip-flop outputs (Q), and Complement Array output ( $\bar{C}$ ). The Complement Array consists of a NOR gate optionally linked to all AND gates for generating and propagating complementary AND terms.

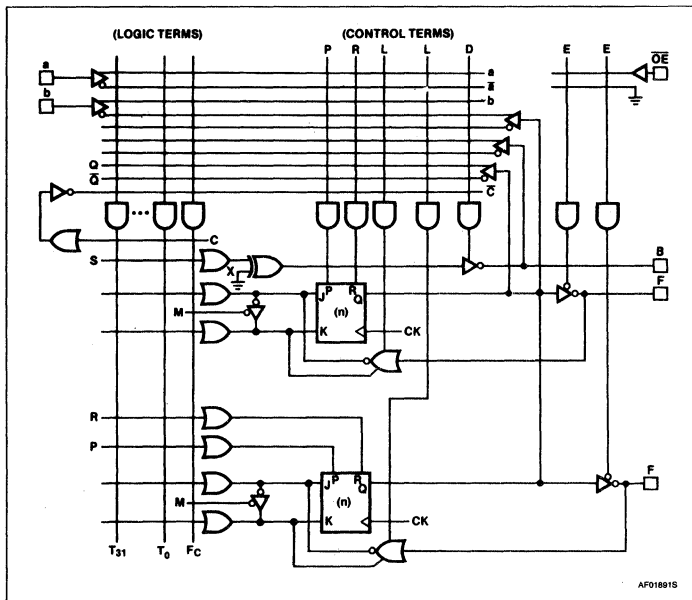
#### FEATURES

- Field-Programmable (NI-Cr link)
- 4 dedicated inputs
- 13 control gates
- 32 AND gates
- 21 OR gates
- 45 product terms:
  - 32 logic terms
  - 13 control terms
- 6 bidirectional I/O lines
- 6 bidirectional registers
- J-K, T, or D-type flip-flops
- Tri-state outputs
- Asynchronous Preset/Reset
- Complement Array
- Active-High or -Low outputs
- Programmable  $\bar{O}E$  control
- Positive edge-triggered clock
- Clock frequency: 15MHz (max.)
- Input loading:  $-100\mu A$  (max.)
- Power dissipation: 750mW (typ.)
- TTL compatible

#### PIN CONFIGURATIONS



#### FUNCTION DIAGRAM



#### APPLICATIONS

- Random sequential logic
- Synchronous up/down counters
- Shift registers
- Bidirectional data buffers
- Timing function generators
- System controllers/synchronizers
- Priority encoder/registers

# Field-Programmable Logic Sequencer (16 × 45 × 12)

PLS157

On-chip T/C buffers couple either True (I, B, Q) or Complement (I,  $\bar{B}$ ,  $\bar{Q}$ ,  $\bar{C}$ ) input polarities to all AND gates, whose outputs can be optionally linked to all OR gates. One group of AND gates drives bidirectional I/O lines (B), whose output polarity is individually programmable through a set of EX-OR gates for implementing AND-OR or AND-NOR logic functions. Another group drives the J-K inputs of all flip-flops. The Asynchronous Preset and Reset lines (P, R), for two of the registers are driven from the AND matrix. The Preset and Reset lines (P, R) controlling the lower four registers are driven from the OR matrix.

All flip-flops are positive edge-triggered and can be used as input, output or I/O (for interfacing with a bidirectional data bus) in conjunction with load control gates (L), steering inputs (I), (B), (Q) and programmable output select lines (E).

The PLS157 is field programmable, enabling the user to quickly generate custom patterns using standard programming equipment.

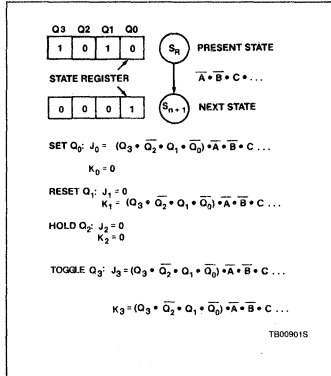
Order codes are contained on the pages following.

### VIRGIN STATE

The factory shipped virgin device contains all fusible links intact, such that:

1.  $\bar{O}\bar{E}$  is always enabled.
2. Preset and Reset are always disabled.
3. All transition terms are disabled.
4. All flip-flops are in D-mode unless otherwise programmed to J/K only or J/K or D (controlled).
5. All B pins are inputs and all F pins are outputs unless otherwise programmed.

### LOGIC FUNCTION



### NOTES:

Similar logic functions are applicable for D and T mode flip-flops.

### FLIP-FLOP TRUTH TABLE

$\bar{O}\bar{E}$	L	C	K	P	R	J	K	Q	F
H									Hi-Z
L	X	X	H	L	X	X		H	L
L	X	X	L	H	X	X		L	H
L	L	$\uparrow$	L	L	L	L		Q	$\bar{Q}$
L	L	$\uparrow$	L	L	L	H		L	H
L	L	$\uparrow$	L	L	H	L		H	L
L	L	$\uparrow$	L	L	H	H		$\bar{Q}$	Q
H	H	$\uparrow$	L	L	L	H		L	H*
H	H	$\uparrow$	L	L	H	L		H	L*
+10V	X	$\uparrow$	X	X	L	H		L	H**
+10V	X	$\uparrow$	X	X	H	L		H	L**

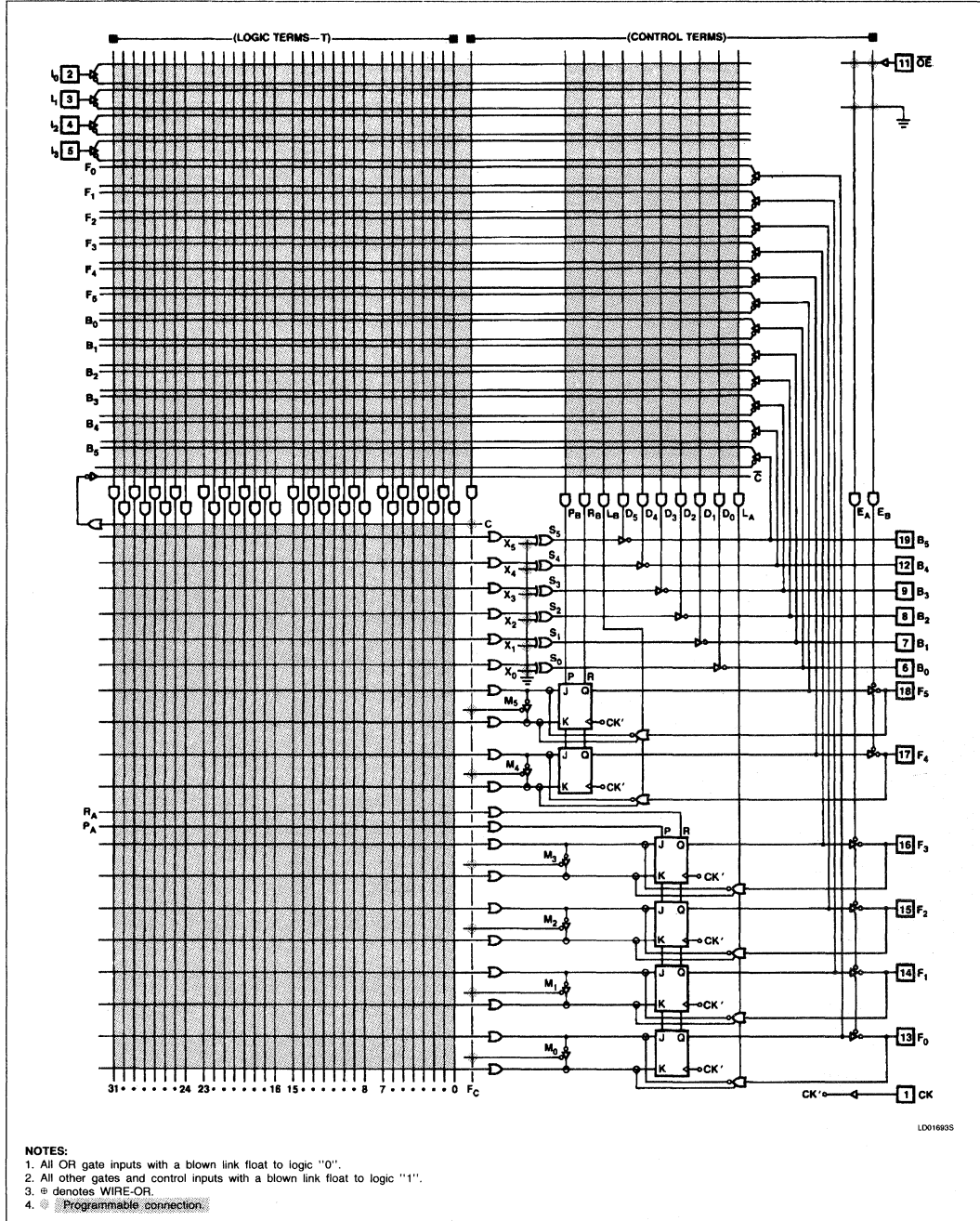
### NOTES:

1. Positive Logic:  
 $J/K = T_0 + T_1 + T_2 \dots T_{31}$   
 $T_n = \bar{C} \cdot (I_0 \cdot I_1 \cdot I_2 \dots) \cdot (Q_0 \cdot Q_1 \dots) \cdot (B_0 \cdot B_1 \dots)$
2.  $\uparrow$  denotes transition from Low to High level.
3. X = Don't care
4. \* = Forced at  $F_n$  pin for loading J/K flip-flop in I/O mode. L must be enabled, and other active  $T_n$  disabled via steering input(s) I, B, or Q.
5. At P = R = H, Q = H. The final state of Q depends on which is released first.
6. \*\* = Forced at  $F_n$  pin to load J/K flip-flop independent of program code (Diagnostic mode), Tri-state B outputs.

# Field-Programmable Logic Sequencer (16 × 45 × 12)

PLS157

## FPLS LOGIC DIAGRAM



- NOTES:**
1. All OR gate inputs with a blown link float to logic "0".
  2. All other gates and control inputs with a blown link float to logic "1".
  3. ⊕ denotes WIRE-OR.
  4. • Programmable connection.

LD016935



## Field-Programmable Logic Sequencer (16 × 45 × 12)

PLS157

## ORDERING INFORMATION

DESCRIPTION	ORDER CODE
20-pin Plastic DIP 300mil-wide	PLS157N
20-pin Plastic Leaded Chip Carrier	PLS157A

## THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS<sup>1</sup>

SYMBOL	PARAMETER	RATING		UNIT
		Min	Max	
V <sub>CC</sub>	Supply voltage		+7	V <sub>DC</sub>
V <sub>IN</sub>	Input voltage		+5.5	V <sub>DC</sub>
V <sub>OUT</sub>	Output voltage		+5.5	V <sub>DC</sub>
I <sub>IN</sub>	Input currents	-30	+30	mA
I <sub>OUT</sub>	Output currents		+100	mA
T <sub>A</sub> T <sub>STG</sub>	Temperature range Operating Storage	0 -65	+75 +150	°C

DC ELECTRICAL CHARACTERISTICS 0°C ≤ T<sub>A</sub> ≤ +75°C, 4.75 ≤ V<sub>CC</sub> ≤ 5.25V

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ <sup>2</sup>	Max	
<b>Input voltage<sup>3</sup></b>						
V <sub>IH</sub> V <sub>IL</sub> V <sub>IC</sub>	High Low Clamp	V <sub>CC</sub> = Max V <sub>CC</sub> = Min V <sub>CC</sub> = Min, I <sub>IN</sub> = -12mA	2		0.8 -1.2	V
<b>Output voltage<sup>3</sup></b>						
V <sub>OH</sub> V <sub>OL</sub>	High Low	V <sub>CC</sub> = Min I <sub>OH</sub> = -2mA I <sub>OL</sub> = 10mA	2.4	0.35	0.5	V
<b>Input current</b>						
I <sub>IH</sub> I <sub>IL</sub>	High Low	V <sub>IN</sub> = 5.5V V <sub>IN</sub> = 0.45V		< 1 -10	80 -100	μA
<b>Output current</b>						
I <sub>O</sub> (OFF) I <sub>OS</sub>	Hi-Z state <sup>8</sup> Short circuit <sup>4</sup>	V <sub>CC</sub> = Max V <sub>OUT</sub> = 5.5V V <sub>OUT</sub> = 0.45V V <sub>OUT</sub> = 0V		1 -1	80 -140 -70	μA mA
I <sub>CC</sub>	V <sub>CC</sub> supply current <sup>7</sup>	V <sub>CC</sub> = Max		150	190	mA
<b>Capacitance</b>						
C <sub>IN</sub> C <sub>OUT</sub>	Input Output	V <sub>CC</sub> = 5.0V V <sub>IN</sub> = 2.0V V <sub>OUT</sub> = 2.0V		8 15		pF

## NOTES:

- Stresses above those listed under "Absolute Maximum Ratings" may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.
- All typical values are at V<sub>CC</sub> = 5V, T<sub>A</sub> = +25°C.
- All voltage values are with respect to network ground terminal.
- Test one at a time.
- Measured with V<sub>IH</sub> applied to  $\bar{O}E$ .
- Duration of short circuit should not exceed 1 second.
- I<sub>CC</sub> is measured with the  $\bar{O}E$  input grounded, all other inputs at 4.5V, and the outputs open.
- Leakage values are a combination of input and output leakage.

# Field-Programmable Logic Sequencer (16 × 45 × 12)

PLS157

**AC ELECTRICAL CHARACTERISTICS**  $0^{\circ}\text{C} \leq T_A \leq +75^{\circ}\text{C}$ ,  $4.75 \leq V_{CC} \leq 5.25\text{V}$ ,  $R_1 = 470\Omega$ ,  $R_2 = 1\text{k}\Omega$

SYMBOL	PARAMETER	TO	FROM	TEST CONDITIONS	LIMITS			UNIT
					Min <sup>5</sup>	Typ <sup>1</sup>	Max	
<b>Pulse width</b>								
t <sub>CKH</sub>	Clock <sup>2</sup> high	CK -	CK +	C <sub>L</sub> = 30pF	25	20	ns	
t <sub>CKL</sub>	Clock low	CK +	CK -		30	20		
t <sub>CKP</sub>	Period	CK +	CK +		70	50		
t <sub>PRH</sub>	Preset/Reset pulse	(I,B) +	(I,B) -		40	30		
<b>Setup time</b>								
t <sub>IS1</sub>	Input	CK +	(I,B) ±	C <sub>L</sub> = 30pF	40	30	ns	
t <sub>IS2</sub>	Input (through F <sub>n</sub> )	CK +	F ±		20	10		
t <sub>IS3</sub>	Input (through Complement Array) <sup>4</sup>	CK +	(I,B) ±		65	40		
<b>Hold time</b>								
t <sub>IH1</sub>	Input	CK +	(I,B) ±	C <sub>L</sub> = 30pF	0	-10	ns	
t <sub>IH2</sub>		CK +	F ±		15	10		
<b>Propagation delays</b>								
t <sub>CKO</sub>	Clock	F ±	CK +	C <sub>L</sub> = 30pF		25	30	ns
t <sub>OE1</sub>	Output enable	F -	$\overline{\text{OE}}$ -	C <sub>L</sub> = 30pF		20	30	
t <sub>OD1</sub>	Output disable <sup>3</sup>	F +	$\overline{\text{OE}}$ +	C <sub>L</sub> = 5pF		20	30	
t <sub>PD</sub>	Output	B ±	(I,B) ±	C <sub>L</sub> = 30pF		40	50	
t <sub>OE2</sub>	Output enable	B ±	(I,B) +	C <sub>L</sub> = 30pF		35	55	
t <sub>OD2</sub>	Output disable <sup>3</sup>	B +	(I,B) -	C <sub>L</sub> = 5pF		30	35	
t <sub>PRO</sub>	Preset/Reset	F ±	(I,B) +	C <sub>L</sub> = 30pF		50	55	

**NOTES:**

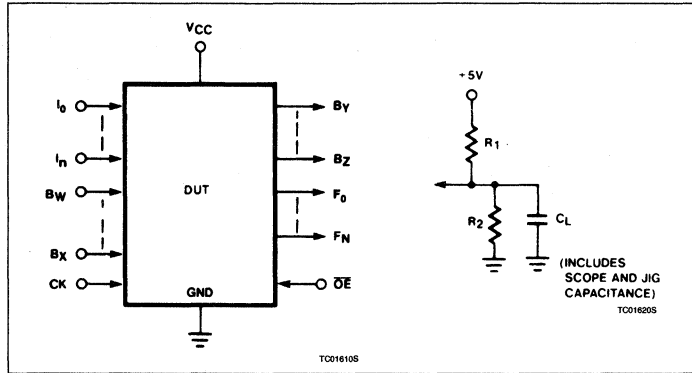
1. All typical values are at  $V_{CC} = 5\text{V}$ ,  $T_A = +25^{\circ}\text{C}$ .
2. To prevent spurious clocking, clock rise time (10% - 90%)  $\leq 10\text{ns}$ .
3. Measured at  $V_T = V_{OL} + 0.5\text{V}$ .
4. When using the Complement Array  $T_{CKP} = 95\text{ns}$  (min).
5. Limits are guaranteed with 12 product terms maximum connected to each sum term line.
6. For test circuits, waveforms and timing diagrams see the following pages.

4

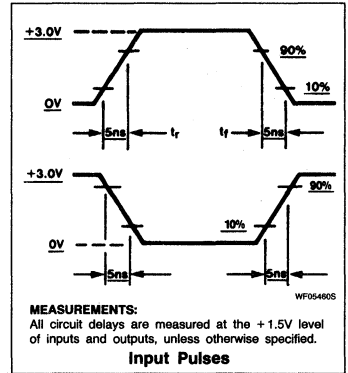
# Field-Programmable Logic Sequencer (16 × 45 × 12)

PLS157

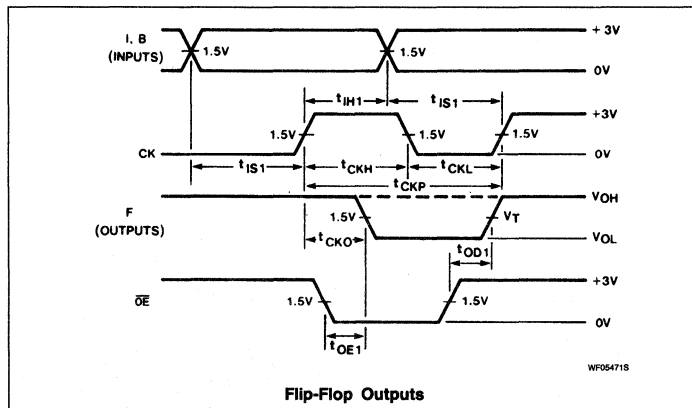
## TEST LOAD CIRCUIT



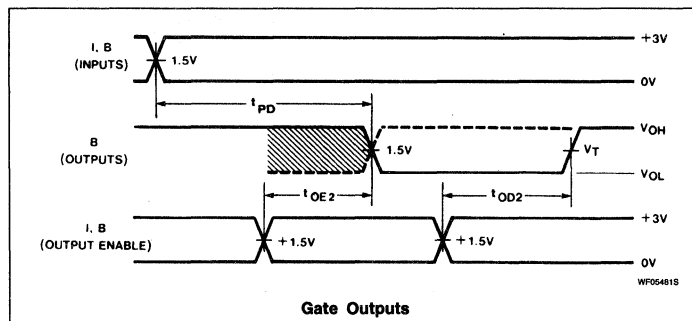
## VOLTAGE WAVEFORM



## TIMING DIAGRAMS



Flip-Flop Outputs



Gate Outputs

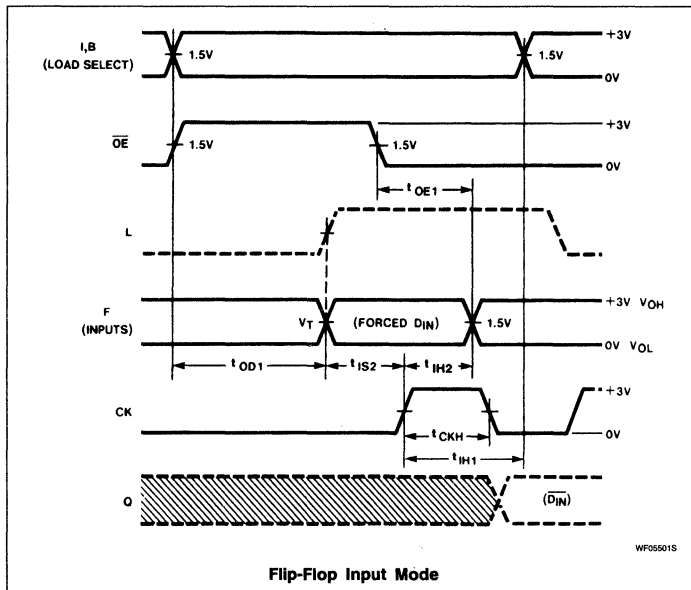
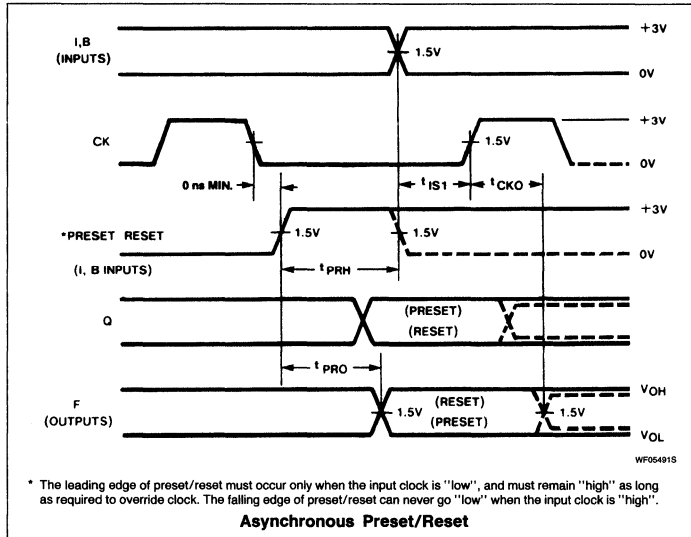
## TIMING DEFINITIONS

- $t_{CKH}$  Width of input clock pulse.
- $t_{CKL}$  Interval between clock pulses.
- $t_{CKP}$  Clock period.
- $t_{PRH}$  Width of preset input pulse.
- $t_{IS1}$  Required delay between beginning of valid input and positive transition of clock.
- $t_{IS2}$  Required delay between beginning of valid input forced at flip-flop output pins, and positive transition of clock.
- $t_{IH1}$  Required delay between positive transition of clock and end of valid input data.
- $t_{IH2}$  Required delay between positive transition of clock and end of valid input data forced at flip-flop output pins.
- $t_{CKO}$  Delay between positive transition of clock and when Outputs become valid (with  $\overline{OE}$  low).
- $t_{OE1}$  Delay between beginning of Output Enable Low and when Outputs become valid.
- $t_{OD1}$  Delay between beginning of Output Enable High and when Outputs are in the off state.
- $t_{PD}$  Propagation delay between combinational inputs and outputs.
- $t_{OE2}$  Delay between predefined Output Enable High, and when combinational Outputs become valid.
- $t_{OD2}$  Delay between predefined Output Enable Low, and when combinational Outputs are in the off state.
- $t_{PRO}$  Delay between positive transition of predefined Preset/Reset input, and when flip-flop outputs become valid.

# Field-Programmable Logic Sequencer (16 × 45 × 12)

PLS157

## TIMING DIAGRAMS (Continued)



# Field-Programmable Logic Sequencer (16 × 45 × 12)

PLS157

The FPLS can be programmed by means of Logic Programming equipment.

With Logic programming, the AND/OR-EX-OR input connections necessary to imple-

ment the desired logic function are coded directly from the State Diagram using the Program Tables on the following pages.

In these tables, the logic state or action of all I/O, control, and state variables is assigned a symbol which results in the proper fusing pattern of corresponding links defined as follows:

## "AND" ARRAY - (I), (B), (Q<sub>p</sub>)

<p>(T, F<sub>C</sub>, L, P, R, D)<sub>n</sub></p> <table border="1"> <thead> <tr> <th>STATE</th> <th>CODE</th> </tr> </thead> <tbody> <tr> <td>INACTIVE<sup>1,2</sup></td> <td>O</td> </tr> </tbody> </table> <p>LS02030S</p>	STATE	CODE	INACTIVE <sup>1,2</sup>	O	<p>(T, F<sub>C</sub>, L, P, R, D)<sub>n</sub></p> <table border="1"> <thead> <tr> <th>STATE</th> <th>CODE</th> </tr> </thead> <tbody> <tr> <td>I, B, Q</td> <td>H</td> </tr> </tbody> </table> <p>LS02040S</p>	STATE	CODE	I, B, Q	H	<p>(T, F<sub>C</sub>, L, P, R, D)<sub>n</sub></p> <table border="1"> <thead> <tr> <th>STATE</th> <th>CODE</th> </tr> </thead> <tbody> <tr> <td>I-bar, B-bar, Q-bar</td> <td>L</td> </tr> </tbody> </table> <p>LS02050S</p>	STATE	CODE	I-bar, B-bar, Q-bar	L	<p>(T, F<sub>C</sub>, L, P, R, D)<sub>n</sub></p> <table border="1"> <thead> <tr> <th>STATE</th> <th>CODE</th> </tr> </thead> <tbody> <tr> <td>DON'T CARE</td> <td>-</td> </tr> </tbody> </table> <p>LS02060S</p>	STATE	CODE	DON'T CARE	-
STATE	CODE																		
INACTIVE <sup>1,2</sup>	O																		
STATE	CODE																		
I, B, Q	H																		
STATE	CODE																		
I-bar, B-bar, Q-bar	L																		
STATE	CODE																		
DON'T CARE	-																		

## "COMPLEMENT" ARRAY - (C)

<p>(T<sub>n</sub>, F<sub>C</sub>)</p> <table border="1"> <thead> <tr> <th>ACTION</th> <th>CODE</th> </tr> </thead> <tbody> <tr> <td>INACTIVE<sup>1,3,5</sup></td> <td>O</td> </tr> </tbody> </table> <p>LS02070S</p>	ACTION	CODE	INACTIVE <sup>1,3,5</sup>	O	<p>(T<sub>n</sub>, F<sub>C</sub>)</p> <table border="1"> <thead> <tr> <th>ACTION</th> <th>CODE</th> </tr> </thead> <tbody> <tr> <td>GENERATE<sup>5</sup></td> <td>A</td> </tr> </tbody> </table> <p>LS02080S</p>	ACTION	CODE	GENERATE <sup>5</sup>	A	<p>(T<sub>n</sub>, F<sub>C</sub>)</p> <table border="1"> <thead> <tr> <th>ACTION</th> <th>CODE</th> </tr> </thead> <tbody> <tr> <td>PROPAGATE</td> <td>•</td> </tr> </tbody> </table> <p>LS02090S</p>	ACTION	CODE	PROPAGATE	•	<p>(T<sub>n</sub>, F<sub>C</sub>)</p> <table border="1"> <thead> <tr> <th>ACTION</th> <th>CODE</th> </tr> </thead> <tbody> <tr> <td>TRANSPARENT</td> <td>-</td> </tr> </tbody> </table> <p>LS02100S</p>	ACTION	CODE	TRANSPARENT	-
ACTION	CODE																		
INACTIVE <sup>1,3,5</sup>	O																		
ACTION	CODE																		
GENERATE <sup>5</sup>	A																		
ACTION	CODE																		
PROPAGATE	•																		
ACTION	CODE																		
TRANSPARENT	-																		

## "OR" ARRAY - (MODE)

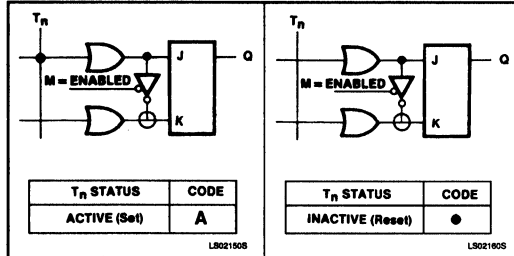
<p>(D<sub>n</sub>, L<sub>n</sub>, P<sub>n</sub>, R<sub>n</sub>)</p> <table border="1"> <thead> <tr> <th>ACTION</th> <th>CODE</th> </tr> </thead> <tbody> <tr> <td>PROPAGATE</td> <td>•</td> </tr> </tbody> </table> <p>LS02110S</p>	ACTION	CODE	PROPAGATE	•	<p>(D<sub>n</sub>, L<sub>n</sub>, P<sub>n</sub>, R<sub>n</sub>)</p> <table border="1"> <thead> <tr> <th>ACTION</th> <th>CODE</th> </tr> </thead> <tbody> <tr> <td>TRANSPARENT</td> <td>-</td> </tr> </tbody> </table> <p>LS02120S</p>	ACTION	CODE	TRANSPARENT	-	<p>(J, K)</p> <table border="1"> <thead> <tr> <th>ACTION</th> <th>CODE</th> </tr> </thead> <tbody> <tr> <td>J-K OR B (CONTROLLED)</td> <td>A</td> </tr> </tbody> </table> <p>LS02130S</p>	ACTION	CODE	J-K OR B (CONTROLLED)	A	<p>(J, K)</p> <table border="1"> <thead> <tr> <th>ACTION</th> <th>CODE</th> </tr> </thead> <tbody> <tr> <td>J-K</td> <td>•</td> </tr> </tbody> </table> <p>LS02140S</p>	ACTION	CODE	J-K	•
ACTION	CODE																		
PROPAGATE	•																		
ACTION	CODE																		
TRANSPARENT	-																		
ACTION	CODE																		
J-K OR B (CONTROLLED)	A																		
ACTION	CODE																		
J-K	•																		

Notes on following page.

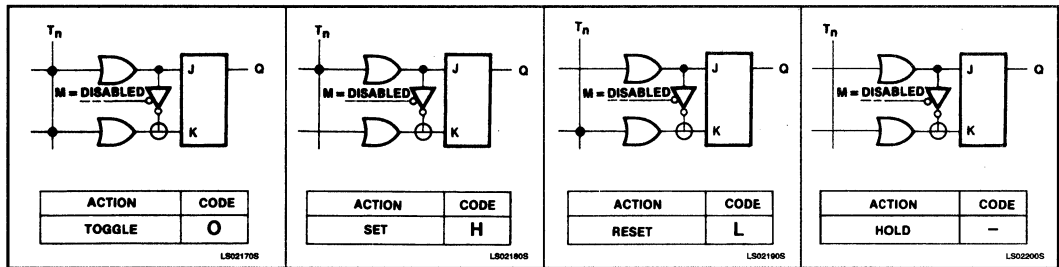
# Field-Programmable Logic Sequencer (16 × 45 × 12)

PLS157

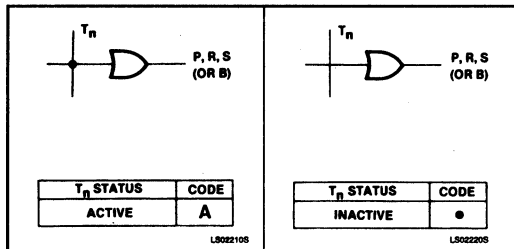
## "OR" ARRAY - ( $Q_n = D$ -Type)



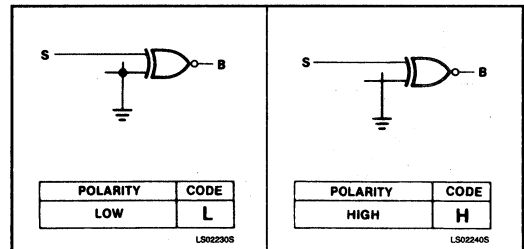
## "OR" ARRAY - ( $Q_n = J$ -K Type)



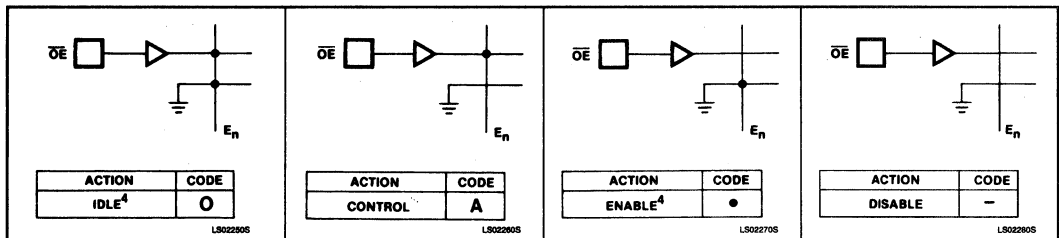
## "OR" ARRAY - (S or B), (P), (R)



## "EX-OR" ARRAY - (B)



## "OE" ARRAY - (E)



**NOTES:**

1. This is the initial unprogrammed state of all link pairs. It is normally associated with all unused (inactive) AND gates.
2. Any gate ( $T_n$ ,  $F_C$ , L, P, R, D)<sub>n</sub> will be unconditionally inhibited if both of the I, B, or Q links are left intact.
3. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates  $T_n$ ,  $F_C$ .
4.  $E_n = 0$  and  $E_n = \bullet$  are logically equivalent states, since both cause  $F_n$  outputs to be unconditionally enabled.
5. These states are not allowed for control gates (L, P, R, D)<sub>n</sub> due to their lack of "OR" array links.



## PLS159 Field-Programmable Logic Sequencer (16 × 45 × 12)

Signetics Programmable Logic  
Product Specification

### Application Specific Products • Series 20

#### DESCRIPTION

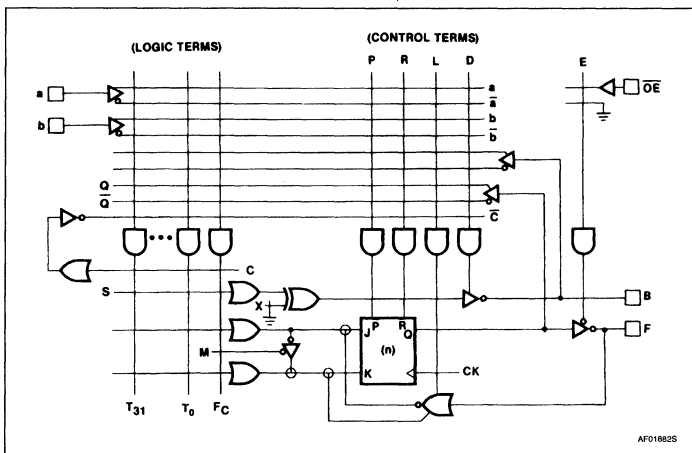
The PLS159 is a Tri-state output, registered logic element combining AND/OR gate arrays with clocked J-K flip-flops. These J-K flip-flops are dynamically convertible to D-type via a "fold-back" inverting buffer and control gate  $F_C$ . It features 8 registered I/O outputs (F) in conjunction with 4 bidirectional I/O lines (B). These yield variable I/O gate and register configurations via control gates (D, L) ranging from 16 inputs to 12 outputs.

The AND/OR arrays consist of 32 logic AND gates, 13 control AND gates, and 21 OR gates with fusible link connections for programming I/O polarity and direction. All AND gates are linked to 4 inputs (I), bidirectional I/O lines (B), internal flip-flop outputs (Q), and Complement Array output ( $\bar{C}$ ). The Complement Array consists of a NOR gate optionally linked to all AND gates for generating and propagating complementary AND terms.

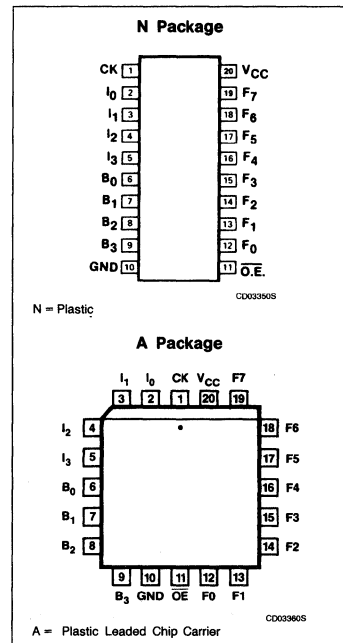
#### FEATURES

- Field-Programmable (Ni-Cr link)
- 4 dedicated inputs
- 13 control gates
- 32 AND gates
- 21 OR gates
- 45 product terms:
  - 32 logic terms
  - 13 control terms
- 4 bidirectional I/O lines
- 8 bidirectional registers
- J/K, T, or D-type flip-flops
- Asynchronous Preset/Reset
- Complement Array
- Active-High or -Low outputs
- Programmable  $\overline{OE}$  control
- Positive edge-triggered clock
- Clock frequency: 15MHz (max.)
- Input loading:  $-100\mu A$  (max.)
- Power dissipation: 750mW (typ.)
- TTL compatible
- Tri-state outputs

#### FUNCTIONAL DIAGRAM



#### PIN CONFIGURATIONS



#### APPLICATIONS

- Random sequential logic
- Synchronous up/down counters
- Shift registers
- Bidirectional data buffers
- Timing function generators
- System controllers/synchronizers
- Priority encoder/registers



# Field-Programmable Logic Sequencer (16 × 45 × 12)

PLS159

On-chip T/C buffers couple either True (I, B, Q) or Complement ( $\bar{I}$ ,  $\bar{B}$ ,  $\bar{Q}$ ,  $\bar{C}$ ) input polarities to all AND gates, whose outputs can be optionally linked to all OR gates. One group of AND gates drives bidirectional I/O lines (B), whose output polarity is individually programmable through a set of EX-OR gates for implementing AND-OR or AND-NOR logic functions. Another group drives the J-K inputs of all flip-flops, as well as asynchronous Preset and Reset lines (P, R).

All flip-flops are positive edge-triggered and can be used as input, output or I/O (for interfacing with a bidirectional data bus) in conjunction with load control gates (L), steering inputs (I), (B), (Q) and programmable output select lines (E).

The PLS159 is field programmable, enabling the user to quickly generate custom patterns using standard programming equipment.

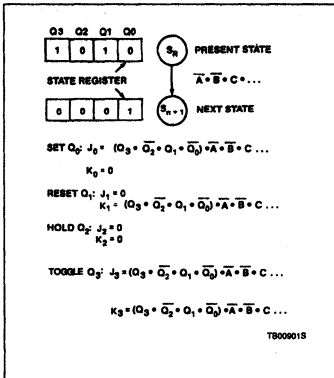
Order codes are contained on the pages following.

### VIRGIN STATE

The factory shipped virgin device contains all fusible links intact, such that:

1.  $\bar{OE}$  is always enabled.
2. Preset and Reset are always disabled.
3. All transition terms are disabled.
4. All flip-flops are in D-mode unless otherwise programmed to J/K only or J/K or D (controlled).
5. All B pins are inputs and all F pins are outputs unless otherwise programmed.

### LOGIC FUNCTION



**NOTE:**  
Similar logic functions are applicable for D and T mode flip-flops.

### FLIP-FLOP TRUTH TABLE

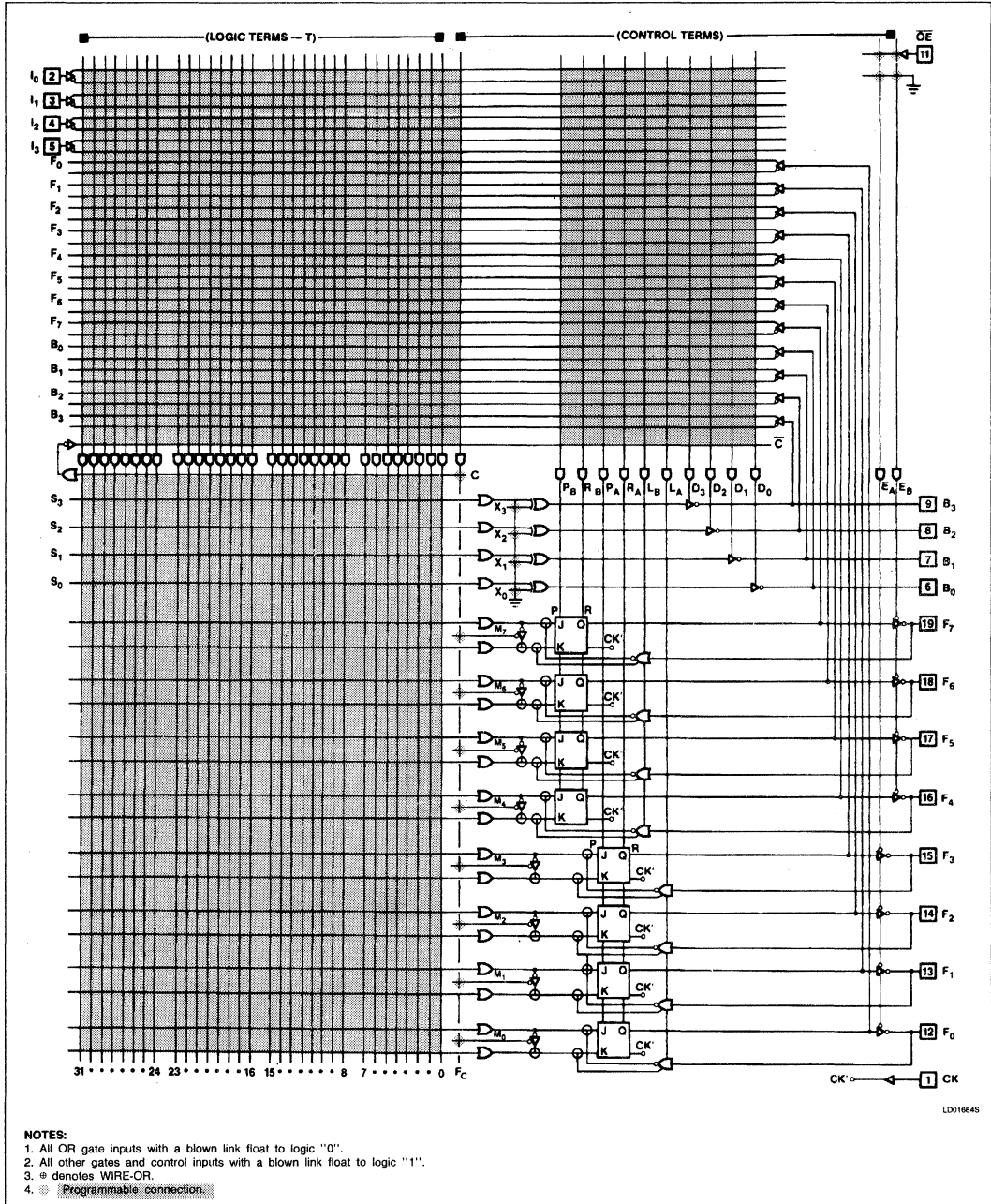
$\bar{OE}$	L	CK	P	R	J	K	Q	F
H								HI-Z
L	X	X	L	X	X	X	L	H
L	X	X	H	L	X	X	H	L
L	X	X	L	H	X	X	L	H
L	L	$\uparrow$	L	L	L	L	Q	$\bar{Q}$
L	L	$\uparrow$	L	L	L	H	L	H
L	L	$\uparrow$	L	L	H	L	H	L
L	L	$\uparrow$	L	L	H	H	$\bar{Q}$	Q
H	H	$\uparrow$	L	L	L	H	L	H*
H	H	$\uparrow$	L	L	H	L	H	L*
+10V	X	$\uparrow$	X	X	L	H	L	H**
	X	$\uparrow$	X	X	H	L	H	L**

- NOTES:**
1. Positive Logic:  
 $J/K = T_0 + T_1 + T_2 \dots T_{31}$   
 $T_n = \bar{C} \cdot (I_0 + I_1 + I_2 \dots) \cdot (Q_0 + Q_1 \dots) \cdot (B_0 + B_1 \dots)$
  2.  $\uparrow$  denotes transition from Low to High level.
  3. X = Don't care
  4. \* = Forced at  $F_n$  pin for loading J/K flip-flop in I/O mode. L must be enabled, and other active  $T_n$  disabled via steering input(s) I, B, or Q.
  5. At P = R = H, Q = H. The final state of Q depends on which is released first.
  6. \*\* = Forced at  $F_n$  pin to load J/K flip-flop independent of program code (Diagnostic mode), Tri-state B outputs.

# Field-Programmable Logic Sequencer (16 × 45 × 12)

PLS159

## FPLS LOGIC DIAGRAM



## Field-Programmable Logic Sequencer (16 × 45 × 12)

PLS159

## ORDERING INFORMATION

DESCRIPTION	ORDER CODE
20-pin Plastic DIP 300mil-wide	PLS159N
20-pin Plastic Leaded Chip Carrier	PLS159A

## THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS<sup>1</sup>

SYMBOL	PARAMETER	RATING		UNIT
		Min	Max	
V <sub>CC</sub>	Supply voltage		+7	V <sub>DC</sub>
V <sub>IN</sub>	Input voltage		+5.5	V <sub>DC</sub>
V <sub>OUT</sub>	Output voltage		+5.5	V <sub>DC</sub>
I <sub>IN</sub>	Input currents	-30	+30	mA
I <sub>OUT</sub>	Output currents		+100	mA
T <sub>A</sub> T <sub>STG</sub>	Temperature range Operating Storage	0 -65	+75 +150	°C

DC ELECTRICAL CHARACTERISTICS 0°C ≤ T<sub>A</sub> ≤ +75°C, 4.75 ≤ V<sub>CC</sub> ≤ 5.25V

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ <sup>2</sup>	Max	
<b>Input voltage<sup>3</sup></b>						
V <sub>IH</sub> V <sub>IL</sub> V <sub>IC</sub>	High Low Clamp	V <sub>CC</sub> = Max V <sub>CC</sub> = Min V <sub>CC</sub> = Min, I <sub>IN</sub> = -12mA	2		0.8 -1.2	V
<b>Output voltage<sup>3</sup></b>						
V <sub>OH</sub> V <sub>OL</sub>	High Low	V <sub>CC</sub> = Min I <sub>OH</sub> = -2mA I <sub>OL</sub> = 10mA	2.4	0.35	0.5	V
<b>Input current<sup>8</sup></b>						
I <sub>IH</sub> I <sub>IL</sub>	High Low	V <sub>CC</sub> = Max V <sub>IN</sub> = 5.5V V <sub>IN</sub> = 0.45V		< 1 -10	80 -100	μA
<b>Output current</b>						
I <sub>O (OFF)</sub> I <sub>OS</sub>	Hi-Z state <sup>8</sup> Short circuit <sup>4</sup>	V <sub>CC</sub> = Max V <sub>OUT</sub> = 5.5V V <sub>OUT</sub> = 0.45V V <sub>OUT</sub> = 0V		1 -1	80 -140 -70	μA mA
I <sub>CC</sub>	V <sub>CC</sub> supply current <sup>7</sup>	V <sub>CC</sub> = Max		150	190	mA
<b>Capacitance</b>						
C <sub>IN</sub> C <sub>OUT</sub>	Input Output	V <sub>CC</sub> = 5.0V V <sub>IN</sub> = 2.0V V <sub>OUT</sub> = 2.0V		8 15		pF

## NOTES:

- Stresses above those listed under "Absolute Maximum Ratings" may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.
- All typical values are at V<sub>CC</sub> = 5V, T<sub>A</sub> = +25°C.
- All voltage values are with respect to network ground terminal.
- Test one at a time.
- Measured with V<sub>IH</sub> applied to  $\overline{OE}$ .
- Duration of short circuit should not exceed 1 second.
- I<sub>CC</sub> is measured with the  $\overline{OE}$  input grounded, all other inputs at 4.5V, and the outputs open.
- Leakage values are a combination of input and output leakage.

# Field-Programmable Logic Sequencer (16 × 45 × 12)

PLS159

## AC ELECTRICAL CHARACTERISTICS $0^{\circ}\text{C} \leq T_A \leq +75^{\circ}\text{C}$ , $4.75 \leq V_{CC} \leq 5.25\text{V}$ , $R_1 = 470\Omega$ , $R_2 = 1\text{k}\Omega$

SYMBOL	PARAMETER	TO	FROM	TEST CONDITIONS	LIMITS			UNIT
					Min <sup>5</sup>	Typ <sup>1</sup>	Max	
<b>Pulse width</b>								
$t_{CKH}$	Clock <sup>2</sup> high	CK -	CK +	$C_L = 30\text{pF}$	25	20		ns
$t_{CKL}$	Clock low	CK +	CK -		30	20		
$t_{CKP}$	Period	CK +	CK +		70	50		
$t_{PRH}$	Preset/Reset pulse	(I,B) +	(I,B) -		40	30		
<b>Setup time</b>								
$t_{IS1}$	Input	CK +	(I,B) $\pm$	$C_L = 30\text{pF}$	40	30		ns
$t_{IS2}$	Input (through $F_n$ )	CK +	F $\pm$		20	10		
$t_{IS3}$	Input (through Complement Array) <sup>4</sup>	CK +	(I,B) $\pm$		65	40		
<b>Hold time</b>								
$t_{IH1}$	Input	CK +	(I,B) $\pm$	$C_L = 30\text{pF}$	0	-10		ns
$t_{IH2}$		CK +	F $\pm$		15	10		
<b>Propagation delays</b>								
$t_{CKO}$	Clock	F $\pm$	CK +	$C_L = 30\text{pF}$		25	30	ns
$t_{OE1}$	Output enable	F -	$\overline{OE}$ -	$C_L = 30\text{pF}$		20	30	
$t_{OD1}$	Output disable <sup>3</sup>	F +	$\overline{OE}$ +	$C_L = 5\text{pF}$		20	35	
$t_{PD}$	Output	B $\pm$	(I,B) $\pm$	$C_L = 30\text{pF}$		40	50	
$t_{OE2}$	Output enable	B $\pm$	(I,B) +	$C_L = 30\text{pF}$		35	55	
$t_{OD2}$	Output disable <sup>3</sup>	B +	(I,B) -	$C_L = 5\text{pF}$		30	35	
$t_{PRO}$	Preset/Reset	F $\pm$	(I,B) +	$C_L = 30\text{pF}$		50	55	

**NOTES:**

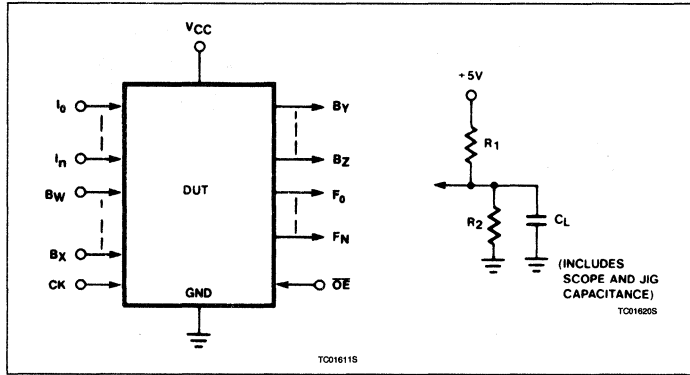
1. All typical values are at  $V_{CC} = 5\text{V}$ ,  $T_A = +25^{\circ}\text{C}$ .
2. To prevent spurious clocking, clock rise time (10% - 90%)  $\leq 10\text{ns}$ .
3. Measured at  $V_I = V_{OL} + 0.5\text{V}$ .
4. When using the Complement Array  $T_{CKP} = 95\text{ns}$  (min).
5. Limits are guaranteed with 12 product terms maximum connected to each sum term line.
6. For test circuits, waveforms and timing diagrams see the following pages.

4

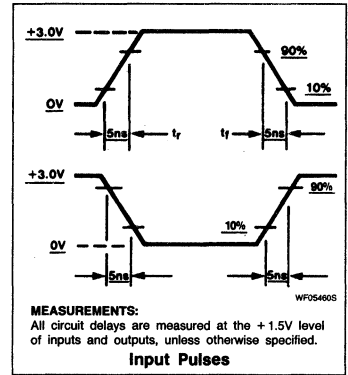
# Field-Programmable Logic Sequencer (16 × 45 × 12)

PLS159

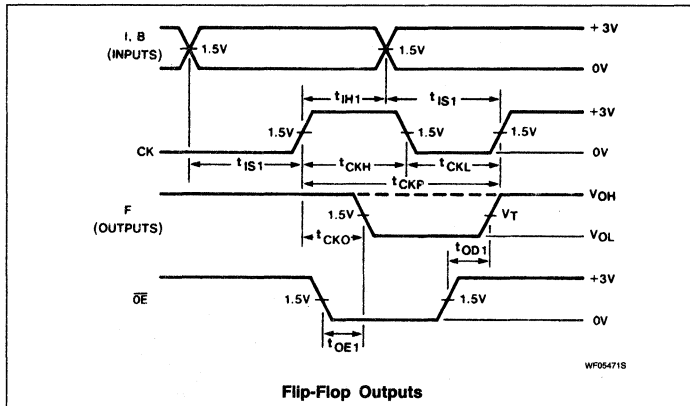
## TEST LOAD CIRCUIT



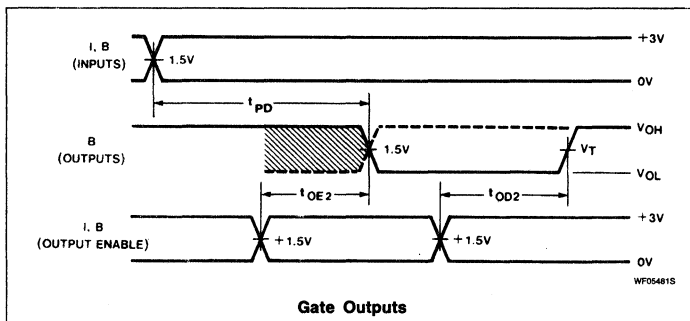
## VOLTAGE WAVEFORM



## TIMING DIAGRAMS



Flip-Flop Outputs



Gate Outputs

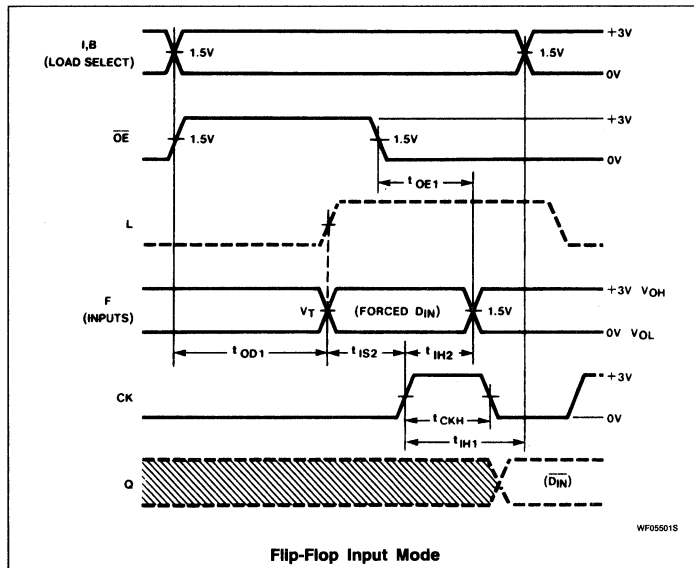
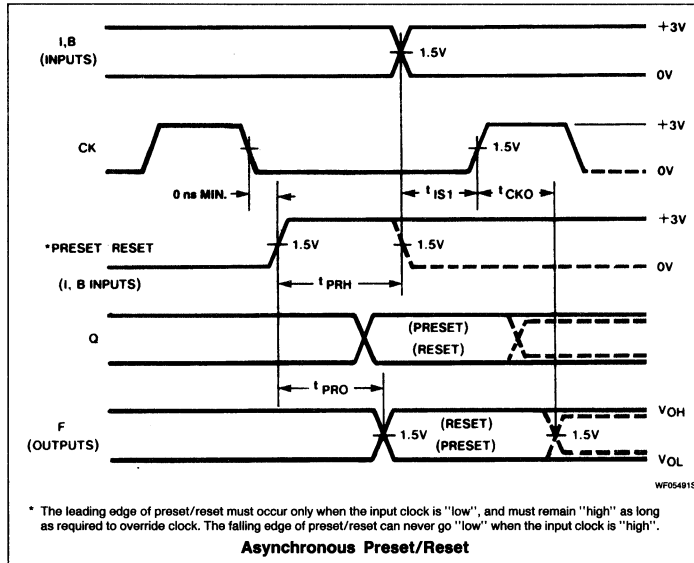
## TIMING DEFINITIONS

- t<sub>CKH</sub> Width of input clock pulse.
- t<sub>CKL</sub> Interval between clock pulses.
- t<sub>CKP</sub> Clock period.
- t<sub>PRH</sub> Width of preset input pulse.
- t<sub>IS1</sub> Required delay between beginning of valid input and positive transition of clock.
- t<sub>IS2</sub> Required delay between beginning of valid input forced at flip-flop output pins, and positive transition of clock.
- t<sub>IH1</sub> Required delay between positive transition of clock and end of valid input data.
- t<sub>IH2</sub> Required delay between positive transition of clock and end of valid input data forced at flip-flop output pins.
- t<sub>CKO</sub> Delay between positive transition of clock and when Outputs become valid (with OE low).
- t<sub>OE1</sub> Delay between beginning of Output Enable Low and when Outputs become valid.
- t<sub>OD1</sub> Delay between beginning of Output Enable High and when Outputs are in the off state.
- t<sub>PD</sub> Propagation delay between combinational inputs and outputs.
- t<sub>OE2</sub> Delay between predefined Output Enable High, and when combinational Outputs become valid.
- t<sub>OD2</sub> Delay between predefined Output Enable Low, and when combinational Outputs are in the off state.
- t<sub>PRO</sub> Delay between positive transition of predefined Preset/Reset input, and when flip-flop outputs become valid.

# Field-Programmable Logic Sequencer (16 × 45 × 12)

PLS159

## TIMING DIAGRAMS (Continued)



4

# Field-Programmable Logic Sequencer (16 × 45 × 12)

PLS159

The FPLS can be programmed by means of Logic Programming equipment.

With Logic programming, the AND/OR-EX-OR input connections necessary to imple-

ment the desired logic function are coded directly from the State Diagram using the Program Tables on the following pages.

In these tables, the logic state or action of all I/O, control, and state variables is assigned a symbol which results in the proper fusing pattern of corresponding links defined as follows:

## "AND" ARRAY - (I), (B), (Q<sub>p</sub>)

<p>(T, F<sub>C</sub>, L, P, R, D)<sub>n</sub></p> <table border="1"> <thead> <tr> <th>STATE</th> <th>CODE</th> </tr> </thead> <tbody> <tr> <td>INACTIVE<sup>1,2</sup></td> <td>O</td> </tr> </tbody> </table> <p>LS020305</p>	STATE	CODE	INACTIVE <sup>1,2</sup>	O	<p>(T, F<sub>C</sub>, L, P, R, D)<sub>n</sub></p> <table border="1"> <thead> <tr> <th>STATE</th> <th>CODE</th> </tr> </thead> <tbody> <tr> <td>I, B, Q</td> <td>H</td> </tr> </tbody> </table> <p>LS020405</p>	STATE	CODE	I, B, Q	H	<p>(T, F<sub>C</sub>, L, P, R, D)<sub>n</sub></p> <table border="1"> <thead> <tr> <th>STATE</th> <th>CODE</th> </tr> </thead> <tbody> <tr> <td>I-bar, B-bar, Q-bar</td> <td>L</td> </tr> </tbody> </table> <p>LS020505</p>	STATE	CODE	I-bar, B-bar, Q-bar	L	<p>(T, F<sub>C</sub>, L, P, R, D)<sub>n</sub></p> <table border="1"> <thead> <tr> <th>STATE</th> <th>CODE</th> </tr> </thead> <tbody> <tr> <td>DON'T CARE</td> <td>-</td> </tr> </tbody> </table> <p>LS020605</p>	STATE	CODE	DON'T CARE	-
STATE	CODE																		
INACTIVE <sup>1,2</sup>	O																		
STATE	CODE																		
I, B, Q	H																		
STATE	CODE																		
I-bar, B-bar, Q-bar	L																		
STATE	CODE																		
DON'T CARE	-																		

## "COMPLEMENT" ARRAY - (C)

<p>(T<sub>n</sub>, F<sub>C</sub>)</p> <table border="1"> <thead> <tr> <th>ACTION</th> <th>CODE</th> </tr> </thead> <tbody> <tr> <td>INACTIVE<sup>1,3,5</sup></td> <td>O</td> </tr> </tbody> </table> <p>LS020705</p>	ACTION	CODE	INACTIVE <sup>1,3,5</sup>	O	<p>(T<sub>n</sub>, F<sub>C</sub>)</p> <table border="1"> <thead> <tr> <th>ACTION</th> <th>CODE</th> </tr> </thead> <tbody> <tr> <td>GENERATE<sup>5</sup></td> <td>A</td> </tr> </tbody> </table> <p>LS020805</p>	ACTION	CODE	GENERATE <sup>5</sup>	A	<p>(T<sub>n</sub>, F<sub>C</sub>)</p> <table border="1"> <thead> <tr> <th>ACTION</th> <th>CODE</th> </tr> </thead> <tbody> <tr> <td>PROPAGATE</td> <td>•</td> </tr> </tbody> </table> <p>LS020905</p>	ACTION	CODE	PROPAGATE	•	<p>(T<sub>n</sub>, F<sub>C</sub>)</p> <table border="1"> <thead> <tr> <th>ACTION</th> <th>CODE</th> </tr> </thead> <tbody> <tr> <td>TRANSPARENT</td> <td>-</td> </tr> </tbody> </table> <p>LS021005</p>	ACTION	CODE	TRANSPARENT	-
ACTION	CODE																		
INACTIVE <sup>1,3,5</sup>	O																		
ACTION	CODE																		
GENERATE <sup>5</sup>	A																		
ACTION	CODE																		
PROPAGATE	•																		
ACTION	CODE																		
TRANSPARENT	-																		

## "OR" ARRAY - (MODE)

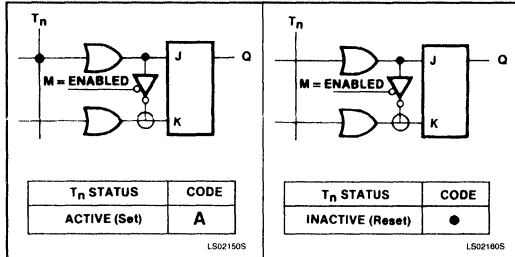
<p>(D<sub>n</sub>, L<sub>n</sub>, P<sub>n</sub>, R<sub>n</sub>)</p> <table border="1"> <thead> <tr> <th>ACTION</th> <th>CODE</th> </tr> </thead> <tbody> <tr> <td>PROPAGATE</td> <td>•</td> </tr> </tbody> </table> <p>LS021105</p>	ACTION	CODE	PROPAGATE	•	<p>(D<sub>n</sub>, L<sub>n</sub>, P<sub>n</sub>, R<sub>n</sub>)</p> <table border="1"> <thead> <tr> <th>ACTION</th> <th>CODE</th> </tr> </thead> <tbody> <tr> <td>TRANSPARENT</td> <td>-</td> </tr> </tbody> </table> <p>LS021205</p>	ACTION	CODE	TRANSPARENT	-	<p>(D<sub>n</sub>, L<sub>n</sub>, P<sub>n</sub>, R<sub>n</sub>)</p> <table border="1"> <thead> <tr> <th>ACTION</th> <th>CODE</th> </tr> </thead> <tbody> <tr> <td>J/K OR D (CONTROLLED)</td> <td>A</td> </tr> </tbody> </table> <p>LS021305</p>	ACTION	CODE	J/K OR D (CONTROLLED)	A	<p>(D<sub>n</sub>, L<sub>n</sub>, P<sub>n</sub>, R<sub>n</sub>)</p> <table border="1"> <thead> <tr> <th>ACTION</th> <th>CODE</th> </tr> </thead> <tbody> <tr> <td>J-K</td> <td>•</td> </tr> </tbody> </table> <p>LS021405</p>	ACTION	CODE	J-K	•
ACTION	CODE																		
PROPAGATE	•																		
ACTION	CODE																		
TRANSPARENT	-																		
ACTION	CODE																		
J/K OR D (CONTROLLED)	A																		
ACTION	CODE																		
J-K	•																		

Notes on following page.

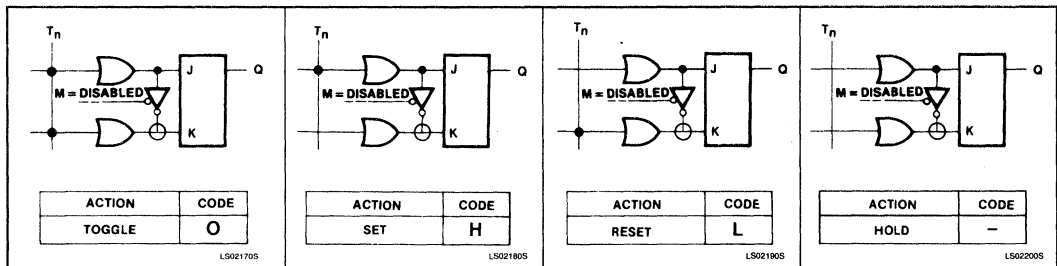
# Field-Programmable Logic Sequencer (16 × 45 × 12)

PLS159

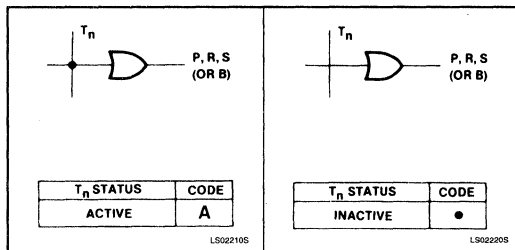
## "OR" ARRAY - ( $Q_n = D$ -Type)



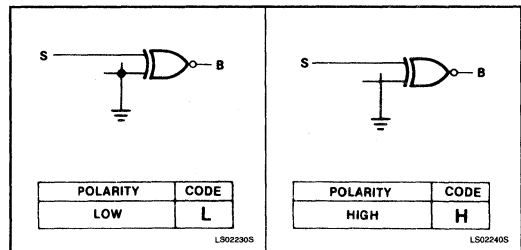
## "OR" ARRAY - ( $Q_n = J$ -K Type)



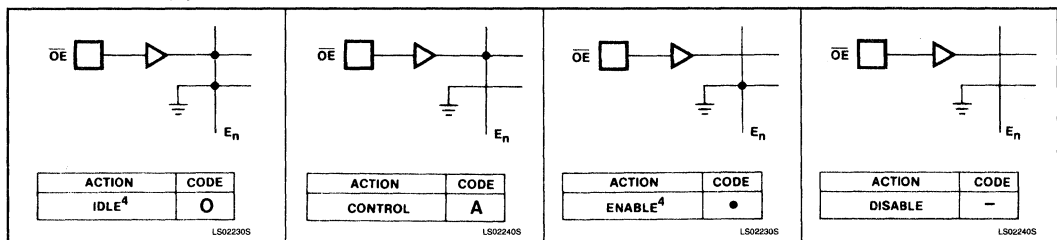
## "OR" ARRAY - (S or B), (P), (R)



## "EX-OR" ARRAY - (B)



## "OE" ARRAY - (E)



**NOTES:**

1. This is the initial unprogrammed state of all link pairs. It is normally associated with all unused (inactive) AND gates.
2. Any gate ( $T_n, F_n, L, P, R, D_n$ ) will be unconditionally inhibited if both of the I, B, or Q links are left intact.
3. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates  $T_n, F_n$ .
4.  $E_n = O$  and  $E_n = \bullet$  are logically equivalent states, since both cause  $F_n$  outputs to be unconditionally enabled.
5. These states are not allowed for control gates ( $L, P, R, D_n$ ) due to their lack of "OR" array links.

4



# Field-Programmable Logic Sequencer (16 × 45 × 12)

PLS159

## FPLS PROGRAM TABLE

AND			OR			CONTROL			NOTES 1. The FPLS is shipped with all links intact. Thus a background of entries corresponding to states of virgin links exists in the table, shown BLANK for clarity. 2. Program unused C, I, B, and Q bits in the AND array as (-). Program unused Q, B, P, and R bits in the OR array as (-) or (A), as applicable. 3. Unused Terms can be left blank. 4. Q (P) and Q (N) are respectively the present and next states of flip-flops Q.																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										
INACTIVE <input type="checkbox"/> O I, B, O <input type="checkbox"/> H T, B, O <input type="checkbox"/> L DON'T CARE -	I, B (I), Q (P)	TOGGLE <input type="checkbox"/> O SET <input type="checkbox"/> H RESET <input type="checkbox"/> L HOLD -	ACTIVE <input type="checkbox"/> A INACTIVE <input type="checkbox"/> *	P, R, B (O), (Q = O)	J/K <input type="checkbox"/> * J/K or D (controlled) <input type="checkbox"/> A	FIF MODE	IDLE <input type="checkbox"/> O CONTROL <input type="checkbox"/> A ENABLE <input type="checkbox"/> * DISABLE -	EA, B																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											
THIS PORTION TO BE COMPLETED BY SIGNETICS  CF (XXXX) _____ CUSTOMER SYMBOLIZED PART # _____ DATE RECEIVED _____ COMMENTS _____			CUSTOMER NAME _____ PURCHASE ORDER # _____ SIGNETICS DEVICE # _____ TOTAL NUMBER OF PARTS _____ PROGRAM TABLE # _____ REV _____ DATE _____			<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th colspan="12" style="text-align: center;">AND</th> <th colspan="12" style="text-align: center;">(OR)</th> </tr> <tr> <th rowspan="2" style="writing-mode: vertical-rl; transform: rotate(180deg);">T E R M</th> <th colspan="12" style="text-align: center;">I</th> <th colspan="12" style="text-align: center;">Q (N)</th> </tr> <tr> <th colspan="3" style="text-align: center;">C</th> <th colspan="3" style="text-align: center;">B(I)</th> <th colspan="6" style="text-align: center;">Q (P)</th> <th colspan="3" style="text-align: center;">B(O)</th> </tr> <tr> <td></td> <td>3</td><td>2</td><td>1</td><td>0</td> <td>3</td><td>2</td><td>1</td><td>0</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> <td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td>0</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>3</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>4</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>5</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>6</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>7</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>8</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>9</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>10</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>11</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>12</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>13</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>14</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>15</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>16</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>17</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>18</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>19</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>20</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>21</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>22</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>23</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>24</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>25</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>26</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>27</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>28</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>29</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>30</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>31</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>FC</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>PB</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>RB</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>LB</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>PA</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>RA</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>LA</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>D3</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>D2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>D1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>D0</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>PN</td><td>5</td><td>4</td><td>3</td><td>2</td><td>9</td><td>8</td><td>7</td><td>6</td><td>10</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> </table>						AND												(OR)												T E R M	I												Q (N)												C			B(I)			Q (P)						B(O)				3	2	1	0	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	3	2	1	0	0																													1																													2																													3																													4																													5																													6																													7																													8																													9																													10																													11																													12																													13																													14																													15																													16																													17																													18																													19																													20																													21																													22																													23																													24																													25																													26																													27																													28																													29																													30																													31																													FC																													PB																													RB																													LB																													PA																													RA																													LA																													D3																													D2																													D1																													D0																													PN	5	4	3	2	9	8	7	6	10	18	17	16	15	14	13	12											
						AND												(OR)																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
T E R M	I												Q (N)																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						
	C			B(I)			Q (P)						B(O)																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						
	3	2	1	0	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	3	2	1	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							
0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
1																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
2																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
3																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
4																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
5																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
6																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
7																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
8																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
9																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
10																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
11																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
12																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
13																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
14																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
15																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
16																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
17																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
18																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
19																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
20																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
21																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
22																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
23																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
24																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
25																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
26																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
27																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
28																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
29																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
30																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
31																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
FC																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
PB																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
RB																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
LB																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
PA																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
RA																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
LA																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
D3																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
D2																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
D1																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
D0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			
PN	5	4	3	2	9	8	7	6	10	18	17	16	15	14	13	12																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																			

## PLS159A Field-Programmable Logic Sequencer (16 × 45 × 12)

Signetics Programmable Logic  
*Preliminary Specification*

### Application Specific Products • Series 20

#### DESCRIPTION

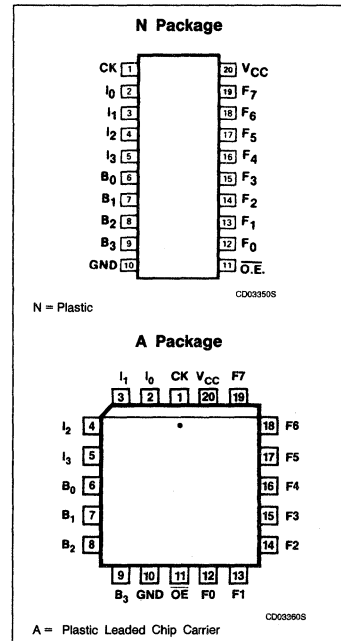
The PLS159A is a Tri-state output, registered logic element combining AND/OR gate arrays with clocked J-K flip-flops. These J-K flip-flops are dynamically convertible to D-type via a "fold-back" inverting buffer and control gate  $F_C$ . It features 8 registered I/O outputs (F) in conjunction with 4 bidirectional I/O lines (B). These yield variable I/O gate and register configurations via control gates (D, L) ranging from 16 inputs to 12 outputs.

The AND/OR arrays consist of 32 logic AND gates, 13 control AND gates, and 21 OR gates with fusible link connections for programming I/O polarity and direction. All AND gates are linked to 4 inputs (I), bidirectional I/O lines (B), internal flip-flop outputs (Q), and Complement Array output ( $\bar{C}$ ). The Complement Array output consists of a NOR gate optionally linked to all AND gates for generating and propagating complementary AND terms.

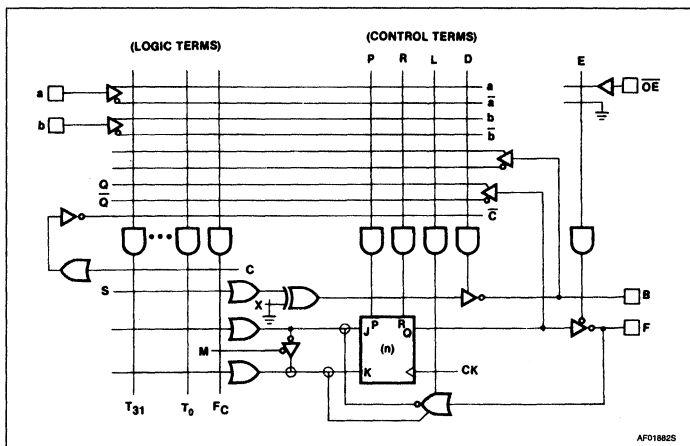
#### FEATURES

- High-speed version of PLS159
- Field-Programmable (Ni-Cr link)
- 4 dedicated inputs
- 13 control gates
- 32 AND gates
- 32 AND gates
- 21 OR gates
- 45 product terms:
  - 32 logic terms
  - 13 control terms
- 4 bidirectional I/O lines
- 8 bidirectional registers
- J/K, T, or D-type flip-flops
- Power on reset feature on all flip-flops ( $F_n = 1$ )
- Asynchronous Preset/Reset
- Complement Array
- Active-High or -Low outputs
- Programmable  $\bar{O}E$  control
- Positive edge-triggered clock
- Clock frequency: 18MHz (max.)
- Input loading:  $-100\mu A$  (max.)
- Power dissipation: 750mW (typ.)
- TTL compatible
- Tri-state outputs

#### PIN CONFIGURATIONS



#### FUNCTIONAL DIAGRAM



#### APPLICATIONS

- Random sequential logic
- Synchronous up/down counters
- Shift registers
- Bidirectional data buffers
- Timing function generators
- System controllers/synchronizers
- Priority encoder/registers

# Field-Programmable Logic Sequencer (16 × 45 × 12)

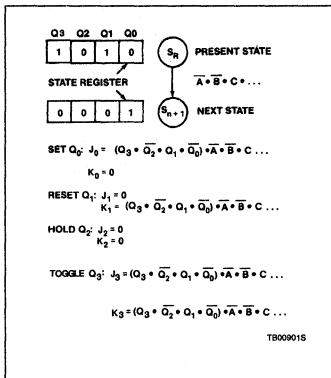
PLS159A

On-chip T/C buffers couple either True (I, B, Q) or Complement (I, B, Q, C) input polarities to all AND gates, whose outputs can be optionally linked to all OR gates. One group of AND gates drives bidirectional I/O lines (B), whose output polarity is individually programmable through a set of EX-OR gates for implementing AND-OR or AND-NOR logic functions. Another group drives the J-K inputs of all flip-flops, as well as asynchronous Preset and Reset lines (P, R).

All flip-flops are positive edge-triggered and can be used as input, output or I/O (for interfacing with a bidirectional data bus) in conjunction with load control gates (L), steering inputs (I), (B), (Q) and programmable output select lines (E).

The PLS159A is field programmable, enabling the user to quickly generate custom patterns using standard programming equipment.

## LOGIC FUNCTION



**NOTE:**  
 Similar logic functions are applicable for D and T mode flip-flops.

## VIRGIN STATE

The factory shipped virgin device contains all fusible links intact, such that:

1.  $\bar{OE}$  is always enabled.
2. Preset and Reset are always disabled.
3. All transition terms are disabled.
4. All flip-flops are in D-mode unless otherwise programmed to J/K only or J/K or D (controlled).
5. All B pins are inputs and all F pins are outputs unless otherwise programmed.

## FLIP-FLOP TRUTH TABLE

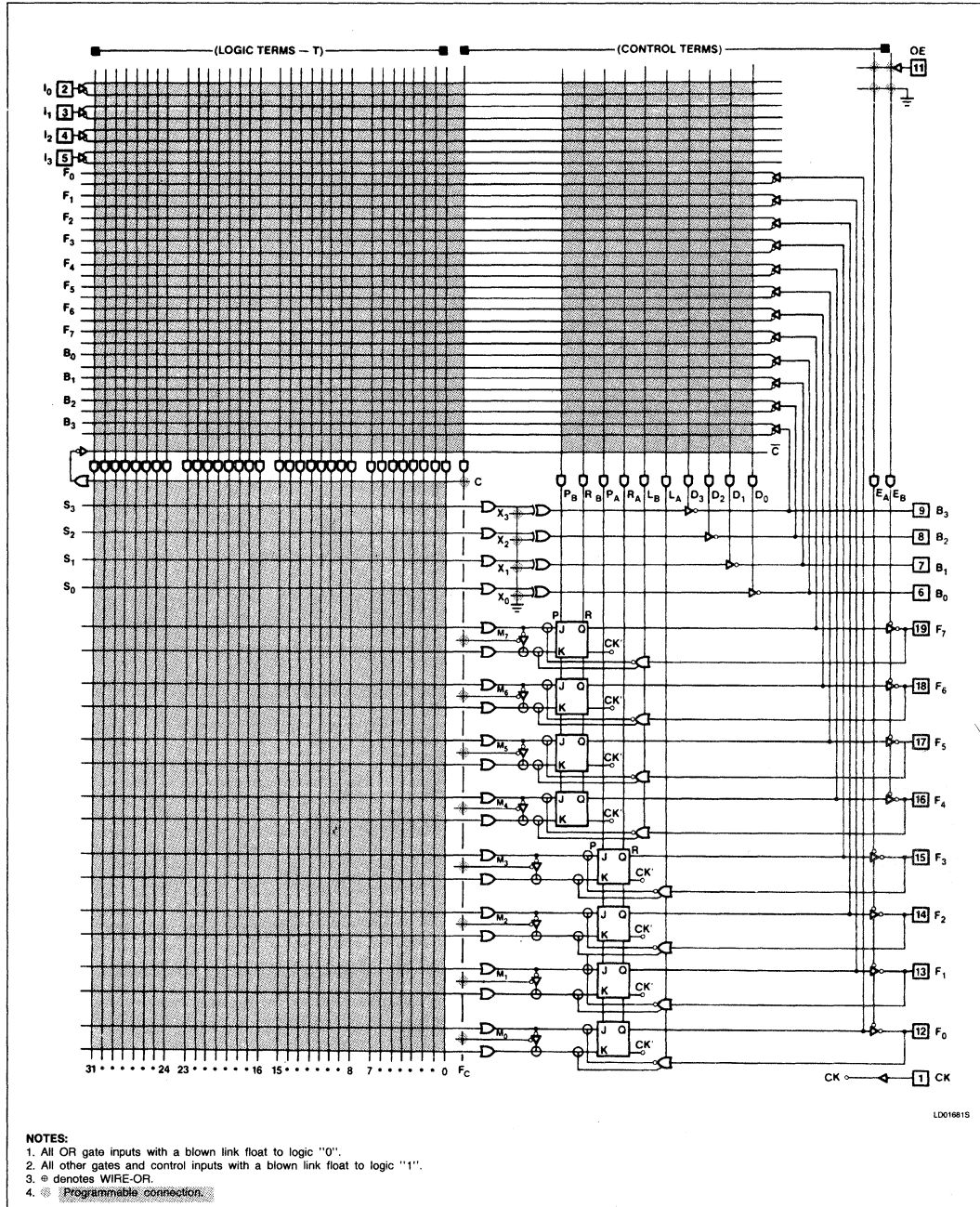
$\bar{OE}$	L	C	K	P	R	J	K	Q	F
H	[Blacked out]							L	Hi-Z
L	X	X	L	X	X	X	X	L	H
L	X	X	H	L	X	X	X	H	L
L	X	X	L	H	X	X	X	L	H
L	L	↑	L	L	L	L	L	Q	$\bar{Q}$
L	L	↑	L	L	L	H	L	L	H
L	L	↑	L	L	H	L	L	H	L
L	L	↑	L	L	H	H	L	$\bar{Q}$	Q
H	H	↑	L	L	L	H	L	L	H*
H	H	↑	L	L	H	L	L	H	L*
+10V	X	↑	X	X	L	H	L	L	H**
	X	↑	X	X	H	L	L	H	L**

- NOTES:**
1. Positive Logic:  
 $J/K = T_0 + T_1 + T_2 \dots T_{31}$   
 $T_n = \bar{C} \cdot (I_0 + I_1 + I_2 \dots) \cdot (Q_0 + Q_1 \dots) \cdot (B_0 + B_1 \dots)$
  2. ↑ denotes transition from Low to High level.
  3. X = Don't Care
  4. \* = Forced at  $F_n$  pin for loading J/K flip-flop in I/O mode. L must be enabled, and other active  $T_n$  disabled via steering input(s) I, B, or Q.
  5. At P = R = H, Q = H. The final state of Q depends on which is released first.
  6. \*\* = Forced at  $F_n$  pin to load J/K flip-flop independent of program code (Diagnostic mode), Tri-state B outputs.

# Field-Programmable Logic Sequencer (16 × 45 × 12)

PLS159A

## FPLS LOGIC DIAGRAM



## Field-Programmable Logic Sequencer (16 × 45 × 12)

PLS159A

## ORDERING INFORMATION

DESCRIPTION	ORDER CODE
20-pin Plastic DIP 300mil-wide	PLS159AN
20-pin Plastic Leaded Chip Carrier	PLS159AA

## THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS<sup>1</sup>

SYMBOL	PARAMETER	RATING		UNIT
		Min	Max	
V <sub>CC</sub>	Supply voltage		+7	V <sub>DC</sub>
V <sub>IN</sub>	Input voltage		+5.5	V <sub>DC</sub>
V <sub>OUT</sub>	Output voltage		+5.5	V <sub>DC</sub>
I <sub>IN</sub>	Input currents	-30	+30	mA
I <sub>OUT</sub>	Output currents		+100	mA
T <sub>A</sub> T <sub>STG</sub>	Temperature range Operating Storage	0 -65	+75 +150	°C

DC ELECTRICAL CHARACTERISTICS 0°C ≤ T<sub>A</sub> ≤ 75°C, 4.75V ≤ V<sub>CC</sub> ≤ 5.25V

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ <sup>2</sup>	Max	
<b>Input voltage<sup>3</sup></b>						
V <sub>IH</sub> V <sub>IL</sub> V <sub>IC</sub>	High Low Clamp	V <sub>CC</sub> = Max V <sub>CC</sub> = Min V <sub>CC</sub> = Min, I <sub>IN</sub> = -12mA	2		0.8 -1.2	V
<b>Output voltage<sup>3</sup></b>						
V <sub>OH</sub> V <sub>OL</sub>	High Low	V <sub>CC</sub> = Min, I <sub>OH</sub> = -2mA I <sub>OL</sub> = 10mA	2.4	0.35	0.5	V
<b>Input current</b>						
I <sub>IH</sub> I <sub>IL</sub>	High Low	V <sub>CC</sub> = Max, V <sub>IN</sub> = 5.5V V <sub>IN</sub> = 0.45V		< 1 -10	40 -100	μA
<b>Output current</b>						
I <sub>O(OFF)</sub>	Hi-Z state <sup>5,8</sup>	V <sub>CC</sub> = Max, V <sub>OUT</sub> = 5.5V V <sub>OUT</sub> = 0.45V		1 -1	80 -140	μA
I <sub>OS</sub>	Short circuit <sup>4,6</sup>	V <sub>OUT</sub> = 0V	-15		-70	mA
I <sub>CC</sub>	V <sub>CC</sub> supply current <sup>7</sup>	V <sub>CC</sub> = Max		150	190	mA
<b>Capacitance</b>						
C <sub>IN</sub> C <sub>OUT</sub>	Input Output	V <sub>CC</sub> = 5.0V, V <sub>IN</sub> = 2.0V V <sub>OUT</sub> = 2.0V		8 15		pF

## NOTES:

- Stresses above those listed under "Absolute Maximum Ratings" may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.
- All typical values are at V<sub>CC</sub> = 5V, T<sub>A</sub> = +25°C.
- All voltage values are with respect to network ground terminal.
- Test one at a time.
- Measured with V<sub>IH</sub> applied to  $\overline{OE}$ .
- Duration of short circuit should not exceed 1 second.
- I<sub>CC</sub> is measured with the  $\overline{OE}$  input grounded, all other inputs at 4.5V, and the outputs open.
- Leakage values are a combination of input and output leakage.

# Field-Programmable Logic Sequencer (16 × 45 × 12)

PLS159A

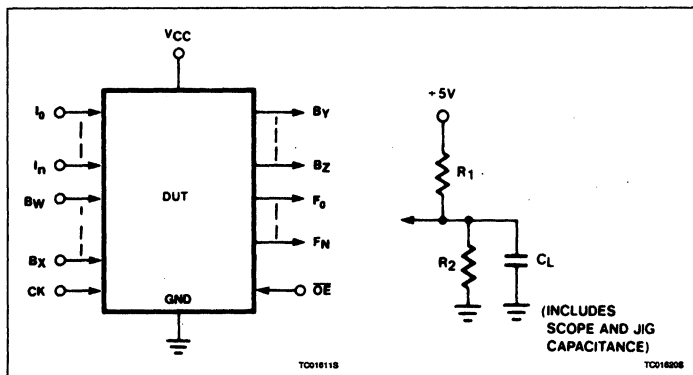
## AC ELECTRICAL CHARACTERISTICS $R_1 = 470\Omega$ , $R_2 = 1k\Omega$ , NPLS178: $0^\circ\text{C} < T_A < +75^\circ\text{C}$ , $4.75\text{V} < V_{CC} < 5.25\text{V}$

SYMBOL	PARAMETER	TO	FROM	TEST CONDITIONS	LIMITS			UNIT
					Min <sup>5</sup>	Typ <sup>1</sup>	Max	
<b>Pulse width</b>								
$t_{CKH}$	Clock <sup>2</sup> high	CK -	CK +	$C_L = 30\text{pF}$	20	15	ns	
$t_{CKL}$	Clock low	CK +	CK -		20	15		
$t_{CKP}$	Period	CK +	CK +		55	45		
$t_{PRH}$	Preset/Reset pulse	(I,B) +	(I,B) -		35	30		
<b>Setup time</b>								
$t_{IS1}$	Input	CK +	(I,B) ±	35	30	ns		
$t_{IS2}$	Input (through $F_n$ )	CK +	F ±	15	10			
$t_{IS3}$	Input (through Complement Array) <sup>4</sup>	CK +	(I,B) ±	55	45			
<b>Hold time</b>								
$t_{IH1}$	Input	CK +	(I,B) ±	0	-5	ns		
$t_{IH2}$	Input (through $F_n$ )	CK +	F ±	15	10			
<b>Propagation delay</b>								
$t_{CKO}$	Clock output enable	F ±	CK ±	$C_L = 5\text{pF}$	15	20	ns	
$t_{OE1}$	Output enable	F -	$\overline{OE}$ -		20	30		
$t_{OD1}$	Output disable <sup>3</sup>	F +	$\overline{OE}$ +	$C_L = 5\text{pF}$	20	30		
$t_{PD}$	Output	B ±	(I,B) ±	$C_L = 30\text{pF}$	25	35		
$t_{OE2}$	Output enable	B ±	(I,B) +		20	30		
$t_{OD2}$	Output disable <sup>3</sup>	B +	(I,B) -	$C_L = 5\text{pF}$	20	30		
$t_{PRO}$	Preset/Reset	F ±	(I,B) +	$C_L = 30\text{pF}$	35	45		
$t_{PPR}$	Power-on preset	F -	$V_{CC}$ +		0	10		

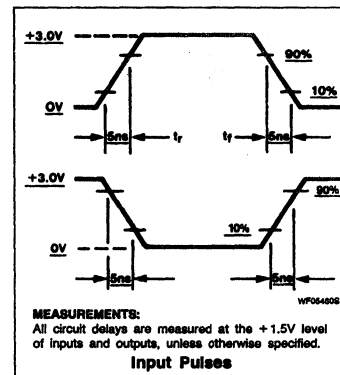
**NOTES:**

1. All typical values are at  $V_{CC} = 5\text{V}$ ,  $T_A = +25^\circ\text{C}$ .
2. To prevent spurious clocking, clock rise time (10% - 90%)  $< 10\text{ns}$ .
3. Measured at  $V_T = V_{OL} + 0.5\text{V}$ .
4. When using the Complement Array  $T_{CKP} = 75\text{ns}$  (min).
5. Limits are guaranteed with 12 product terms maximum connected to each sum term line.

**TEST LOAD CIRCUIT**



**VOLTAGE WAVEFORM**

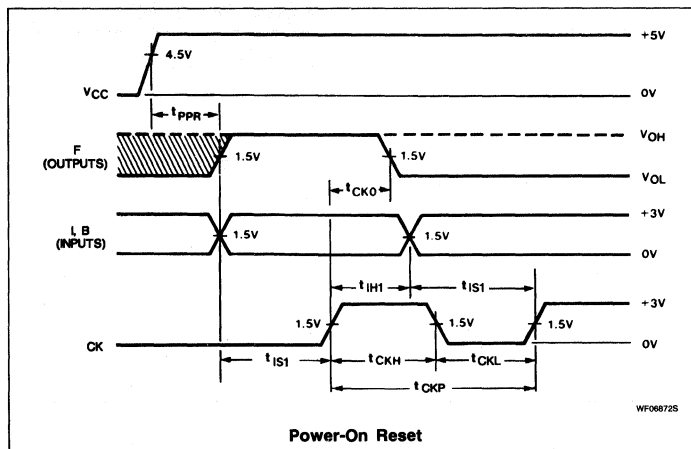
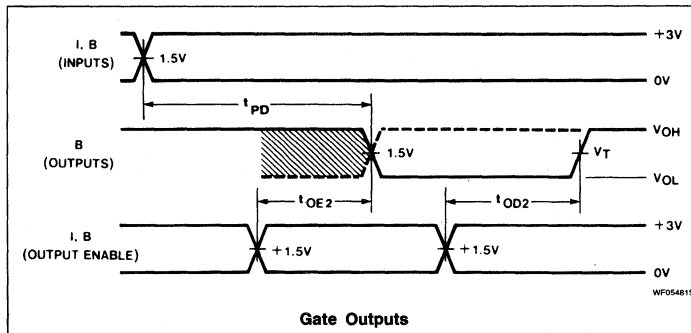
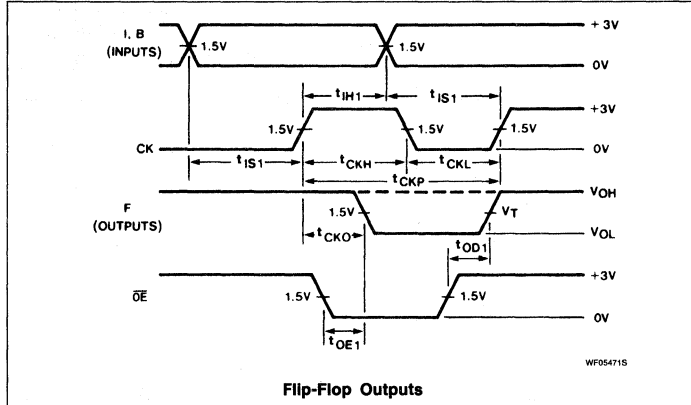


4

# Field-Programmable Logic Sequencer (16 × 45 × 12)

## PLS159A

### TIMING DIAGRAMS



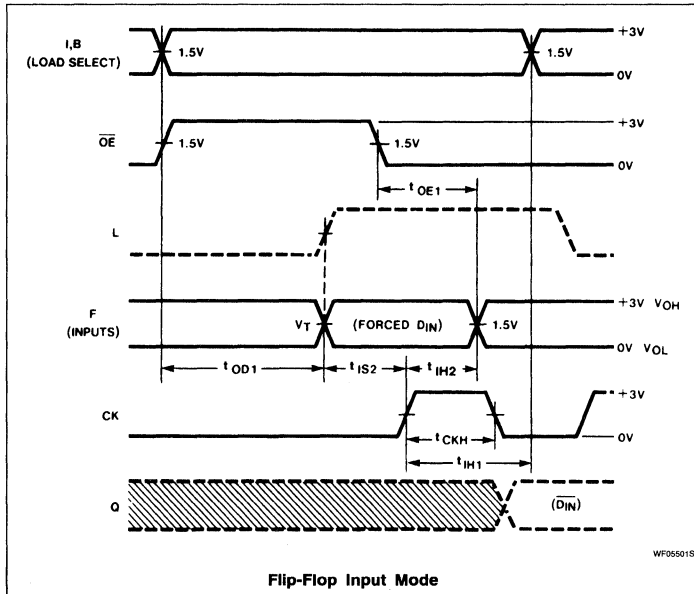
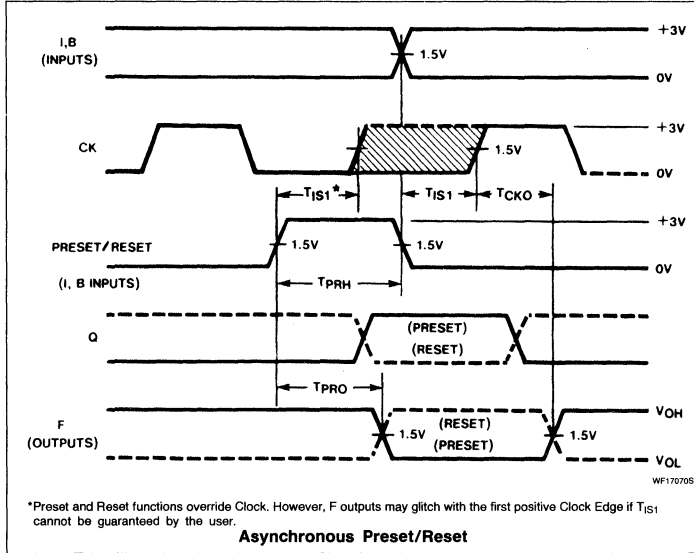
### TIMING DEFINITIONS

- $t_{CKH}$  Width of input clock pulse.
- $t_{CKL}$  Interval between clock pulses.
- $t_{CKP}$  Clock period.
- $t_{PRH}$  Width of preset input pulse.
- $t_{IS1}$  Required delay between beginning of valid input and positive transition of clock.
- $t_{IS2}$  Required delay between beginning of valid input forced at flip-flop output pins, and positive transition of clock.
- $t_{IH1}$  Required delay between positive transition of clock and end of valid input data.
- $t_{IH2}$  Required delay between positive transition of clock and end of valid input data forced at flip-flop output pins.
- $t_{CKO}$  Delay between positive transition of clock and when Outputs become valid (with  $\overline{OE}$  low).
- $t_{OE1}$  Delay between beginning of Output Enable Low and when Outputs become valid.
- $t_{OD1}$  Delay between beginning of Output Enable High and when Outputs are in the off state.
- $t_{PPR}$  Delay between  $V_{CC}$  (after power-on) and when flip-flop outputs become preset at "1" (internal Q outputs at "0").
- $t_{PD}$  Propagation delay between combinational inputs and outputs.
- $t_{OE2}$  Delay between predefined Output Enable High, and when combinational Outputs become valid.
- $t_{OD2}$  Delay between predefined Output Enable Low and when combinational Outputs are in the off state.
- $t_{PRO}$  Delay between positive transition of predefined Preset/Reset input, and when flip-flop outputs become valid.

Field-Programmable Logic Sequencer (16 × 45 × 12)

PLS159A

TIMING DIAGRAMS (Continued)





# Field-Programmable Logic Sequencer (16 × 45 × 12)

PLS159A

The FPLS can be programmed by means of Logic Programming equipment.

With Logic programming, the AND/OR-EX-OR input connections necessary to imple-

ment the desired logic function are coded directly from the State Diagram using the Program Tables on the following pages.

In these Tables, the logic state or action of all I/O, control, and state variables is assigned a symbol which results in the proper fusing pattern of corresponding links defined as follows:

## "AND" ARRAY - (I), (B), (Qp)

<p>(T, F<sub>C</sub>, L, P, R, D)<sub>n</sub></p> <table border="1"> <thead> <tr> <th>STATE</th> <th>CODE</th> </tr> </thead> <tbody> <tr> <td>INACTIVE<sup>1,2</sup></td> <td>O</td> </tr> </tbody> </table> <p>LS020305</p>	STATE	CODE	INACTIVE <sup>1,2</sup>	O	<p>(T, F<sub>C</sub>, L, P, R, D)<sub>n</sub></p> <table border="1"> <thead> <tr> <th>STATE</th> <th>CODE</th> </tr> </thead> <tbody> <tr> <td>I, B, Q</td> <td>H</td> </tr> </tbody> </table> <p>LS020405</p>	STATE	CODE	I, B, Q	H	<p>(T, F<sub>C</sub>, L, P, R, D)<sub>n</sub></p> <table border="1"> <thead> <tr> <th>STATE</th> <th>CODE</th> </tr> </thead> <tbody> <tr> <td>I, B, Q</td> <td>L</td> </tr> </tbody> </table> <p>LS020505</p>	STATE	CODE	I, B, Q	L	<p>(T, F<sub>C</sub>, L, P, R, D)<sub>n</sub></p> <table border="1"> <thead> <tr> <th>STATE</th> <th>CODE</th> </tr> </thead> <tbody> <tr> <td>DON'T CARE</td> <td>-</td> </tr> </tbody> </table> <p>LS020605</p>	STATE	CODE	DON'T CARE	-
STATE	CODE																		
INACTIVE <sup>1,2</sup>	O																		
STATE	CODE																		
I, B, Q	H																		
STATE	CODE																		
I, B, Q	L																		
STATE	CODE																		
DON'T CARE	-																		

## "COMPLEMENT" ARRAY - (C)

<p>(T<sub>n</sub>, F<sub>C</sub>)</p> <table border="1"> <thead> <tr> <th>ACTION</th> <th>CODE</th> </tr> </thead> <tbody> <tr> <td>INACTIVE<sup>1,3,5</sup></td> <td>O</td> </tr> </tbody> </table> <p>LS020705</p>	ACTION	CODE	INACTIVE <sup>1,3,5</sup>	O	<p>(T<sub>n</sub>, F<sub>C</sub>)</p> <table border="1"> <thead> <tr> <th>ACTION</th> <th>CODE</th> </tr> </thead> <tbody> <tr> <td>GENERATE<sup>5</sup></td> <td>A</td> </tr> </tbody> </table> <p>LS020805</p>	ACTION	CODE	GENERATE <sup>5</sup>	A	<p>(T<sub>n</sub>, F<sub>C</sub>)</p> <table border="1"> <thead> <tr> <th>ACTION</th> <th>CODE</th> </tr> </thead> <tbody> <tr> <td>PROPAGATE</td> <td>•</td> </tr> </tbody> </table> <p>LS020905</p>	ACTION	CODE	PROPAGATE	•	<p>(T<sub>n</sub>, F<sub>C</sub>)</p> <table border="1"> <thead> <tr> <th>ACTION</th> <th>CODE</th> </tr> </thead> <tbody> <tr> <td>TRANSPARENT</td> <td>-</td> </tr> </tbody> </table> <p>LS021005</p>	ACTION	CODE	TRANSPARENT	-
ACTION	CODE																		
INACTIVE <sup>1,3,5</sup>	O																		
ACTION	CODE																		
GENERATE <sup>5</sup>	A																		
ACTION	CODE																		
PROPAGATE	•																		
ACTION	CODE																		
TRANSPARENT	-																		

## "OR" ARRAY - (MODE)

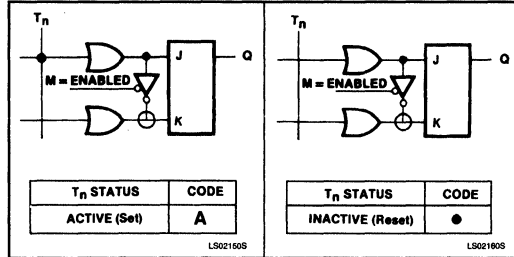
<p>(D<sub>n</sub>, L<sub>n</sub>, P<sub>n</sub>, R<sub>n</sub>)</p> <table border="1"> <thead> <tr> <th>ACTION</th> <th>CODE</th> </tr> </thead> <tbody> <tr> <td>PROPAGATE</td> <td>•</td> </tr> </tbody> </table> <p>LS021105</p>	ACTION	CODE	PROPAGATE	•	<p>(D<sub>n</sub>, L<sub>n</sub>, P<sub>n</sub>, R<sub>n</sub>)</p> <table border="1"> <thead> <tr> <th>ACTION</th> <th>CODE</th> </tr> </thead> <tbody> <tr> <td>TRANSPARENT</td> <td>-</td> </tr> </tbody> </table> <p>LS021205</p>	ACTION	CODE	TRANSPARENT	-	<p>(D<sub>n</sub>, L<sub>n</sub>, P<sub>n</sub>, R<sub>n</sub>)</p> <table border="1"> <thead> <tr> <th>ACTION</th> <th>CODE</th> </tr> </thead> <tbody> <tr> <td>J/K OR D (CONTROLLED)</td> <td>A</td> </tr> </tbody> </table> <p>LS021305</p>	ACTION	CODE	J/K OR D (CONTROLLED)	A	<p>(D<sub>n</sub>, L<sub>n</sub>, P<sub>n</sub>, R<sub>n</sub>)</p> <table border="1"> <thead> <tr> <th>ACTION</th> <th>CODE</th> </tr> </thead> <tbody> <tr> <td>J-K</td> <td>•</td> </tr> </tbody> </table> <p>LS021405</p>	ACTION	CODE	J-K	•
ACTION	CODE																		
PROPAGATE	•																		
ACTION	CODE																		
TRANSPARENT	-																		
ACTION	CODE																		
J/K OR D (CONTROLLED)	A																		
ACTION	CODE																		
J-K	•																		

Notes on following page.

# Field-Programmable Logic Sequencer (16 × 45 × 12)

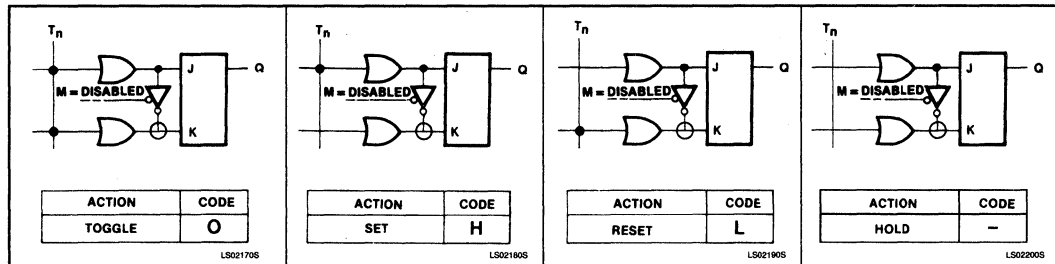
PLS159A

## "OR" ARRAY — ( $Q_N = D$ - Type)

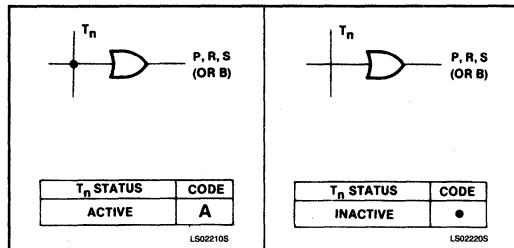


**CAUTION:**  
The PLS159A Programming Algorithm is different from the PLS159.

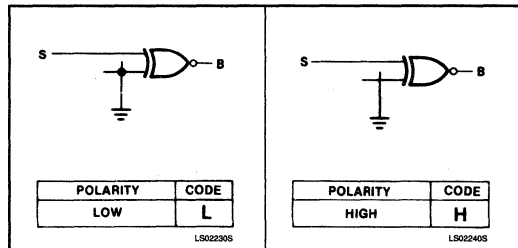
## "AND" ARRAY — ( $Q_N = J - K$ Type)



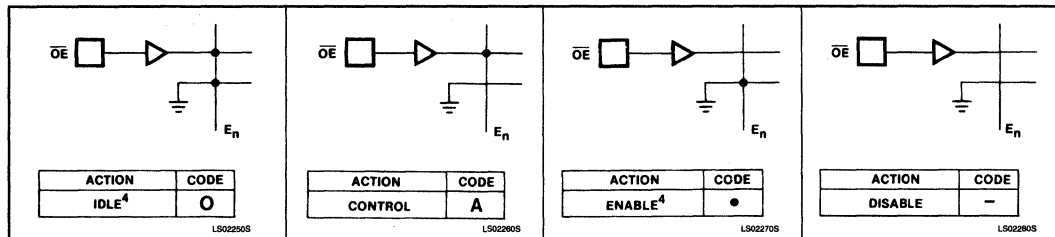
## "OR" ARRAY — (S or B),(P),(R)



## "EX-OR" ARRAY — (B)



## "OE" ARRAY — (E)



- NOTES:**
- This is the initial unprogrammed state of all link pairs. It is normally associated with all unused (inactive) AND gates.
  - Any gate (T, F<sub>C</sub>, L, P, R, D)<sub>n</sub> will be unconditionally inhibited if both of the I, B, or Q links are left intact.
  - To prevent oscillations, this state is not allowed for C link, pairs coupled to active gates T<sub>n</sub>, F<sub>C</sub>.
  - E<sub>n</sub> = O and E<sub>n</sub> = • are logically equivalent states, since both cause F<sub>n</sub> outputs to be unconditionally enabled.
  - These states are not allowed for control gates (L, P, R, D)<sub>n</sub> due to their lack of "OR" array links.

# Field-Programmable Logic Sequencer (16 × 45 × 12)

PLS159A

## FPLS PROGRAM TABLE

AND				OR				CONTROL				NOTES			
INACTIVE <input type="checkbox"/> O	L, B, O <input type="checkbox"/> H	T, B, O <input type="checkbox"/> L	DON'T CARE <input type="checkbox"/> -	ACTIVE <input type="checkbox"/> A	INACTIVE <input type="checkbox"/> B	P, R, B (O), (Q = D)		J/K <input type="checkbox"/> *	J/K or D (controlled) <input type="checkbox"/> A	F/F MODE	IDLE <input type="checkbox"/> O	CONTROL <input type="checkbox"/> A	ENABLE <input type="checkbox"/> *	DISABLE <input type="checkbox"/> -	
I, B (O), Q (P)				TOGGLE <input type="checkbox"/> O				HIGH <input type="checkbox"/> H		(POL.)		E, A, B			
INACTIVE <input type="checkbox"/> O				SET <input type="checkbox"/> H				LOW <input type="checkbox"/> L							
GENERATE <input type="checkbox"/> A				RESET <input type="checkbox"/> L											
PROPAGATE <input type="checkbox"/> *				HOLD <input type="checkbox"/> -											
TRANSPARENT <input type="checkbox"/> -															

T E R M C	AND										(OR)										POLARITY						
	I					B(I)					Q (P)					Q (N)						B(O)					
	3	2	1	0	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4		3	2	1	0	3	2
0																											
1																											
2																											
3																											
4																											
5																											
6																											
7																											
8																											
9																											
10																											
11																											
12																											
13																											
14																											
15																											
16																											
17																											
18																											
19																											
20																											
21																											
22																											
23																											
24																											
25																											
26																											
27																											
28																											
29																											
30																											
31																											
F <sub>C</sub>																											
P <sub>B</sub>																											
R <sub>B</sub>																											
L <sub>B</sub>																											
P <sub>A</sub>																											
R <sub>A</sub>																											
L <sub>A</sub>																											
D <sub>3</sub>																											
D <sub>2</sub>																											
D <sub>1</sub>																											
D <sub>0</sub>																											
PN	5	4	3	2	9	8	7	6	10	18	17	16	15	14	13	12											

THIS PORTION TO BE COMPLETED BY SIGNETICS

CF (XXXX) \_\_\_\_\_

CUSTOMER SYMBOLIZED PART # \_\_\_\_\_

DATE RECEIVED \_\_\_\_\_

COMMENTS \_\_\_\_\_

CUSTOMER NAME \_\_\_\_\_

PURCHASE ORDER # \_\_\_\_\_

SIGNETICS DEVICE # \_\_\_\_\_

TOTAL NUMBER OF PARTS \_\_\_\_\_

PROGRAM TABLE # \_\_\_\_\_ REV \_\_\_\_\_ DATE \_\_\_\_\_

TB01291S

## PLHS18P8A Programmable AND Array Logic (18 × 72 × 8)

Signetics Programmable Logic  
Product Specification

### Application Specific Products • Series 20

#### DESCRIPTION

The PLHS18P8A is a two-level logic element consisting of 72 AND gates and 8 OR gates with fusible connections for programming I/O polarity and direction.

All AND gates are linked to 10 inputs (I) and 8 bidirectional I/O lines (B). These yield variable I/O gate configurations via 8 direction control gates, ranging from 18 inputs to 8 outputs.

On-chip T/C buffers couple either True (I, B) or Complement (I, B) input polarities to all AND gates. The 72 AND gates are separated into 8 groups of 9 each. Each group of 9 is associated with one bidirectional pin. In each group, eight of the AND terms are ORed together, while the ninth is used to establish I/O direction. All outputs are individually programmable via an EX-OR gate to allow implementation of AND/OR or NAND/NOR logic functions.

In the virgin state, the AND array fuses are back-to-back CB-EB diode pairs which will act as open connections. Current is avalanched across individual diode pairs during fusing, which essentially short circuits the EB diode and provides the connection for the associated product term.

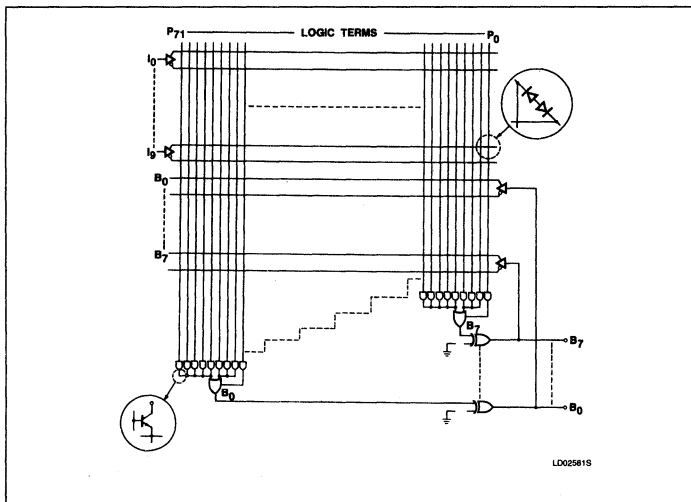
The PLHS18P8A is field-programmable, allowing the user to quickly generate custom pattern using standard programming equipment.

Order codes are contained in the pages following.

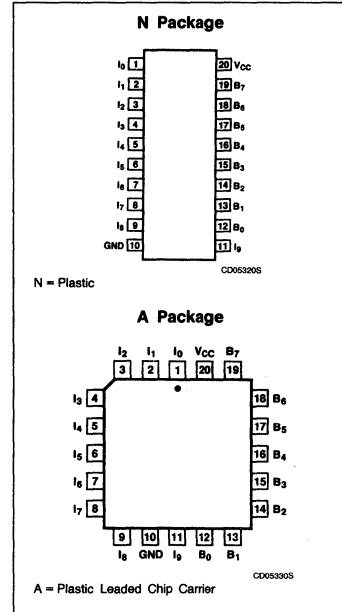
#### FEATURES

- 100% functionally compatible with AmpAL18P8A
- Field-Programmable
- 10 Inputs
- 8 bidirectional I/O lines
- 72 AND gates/product terms - configured into eight groups of nine
- Programmable output polarity (Tri-state output)
- I/O propagation delay: 20ns (max.)
- Power dissipation: 750mW (nominal)
- TTL compatible
- Verify Lock Fuse
- On-chip test features for extensive AC and DC parametric testing

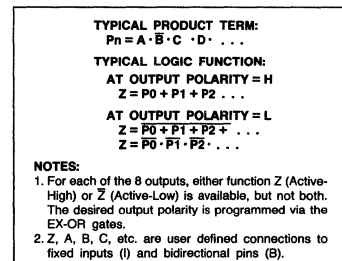
#### FUNCTIONAL DIAGRAM



#### PIN CONFIGURATIONS



#### LOGIC FUNCTION



#### APPLICATIONS

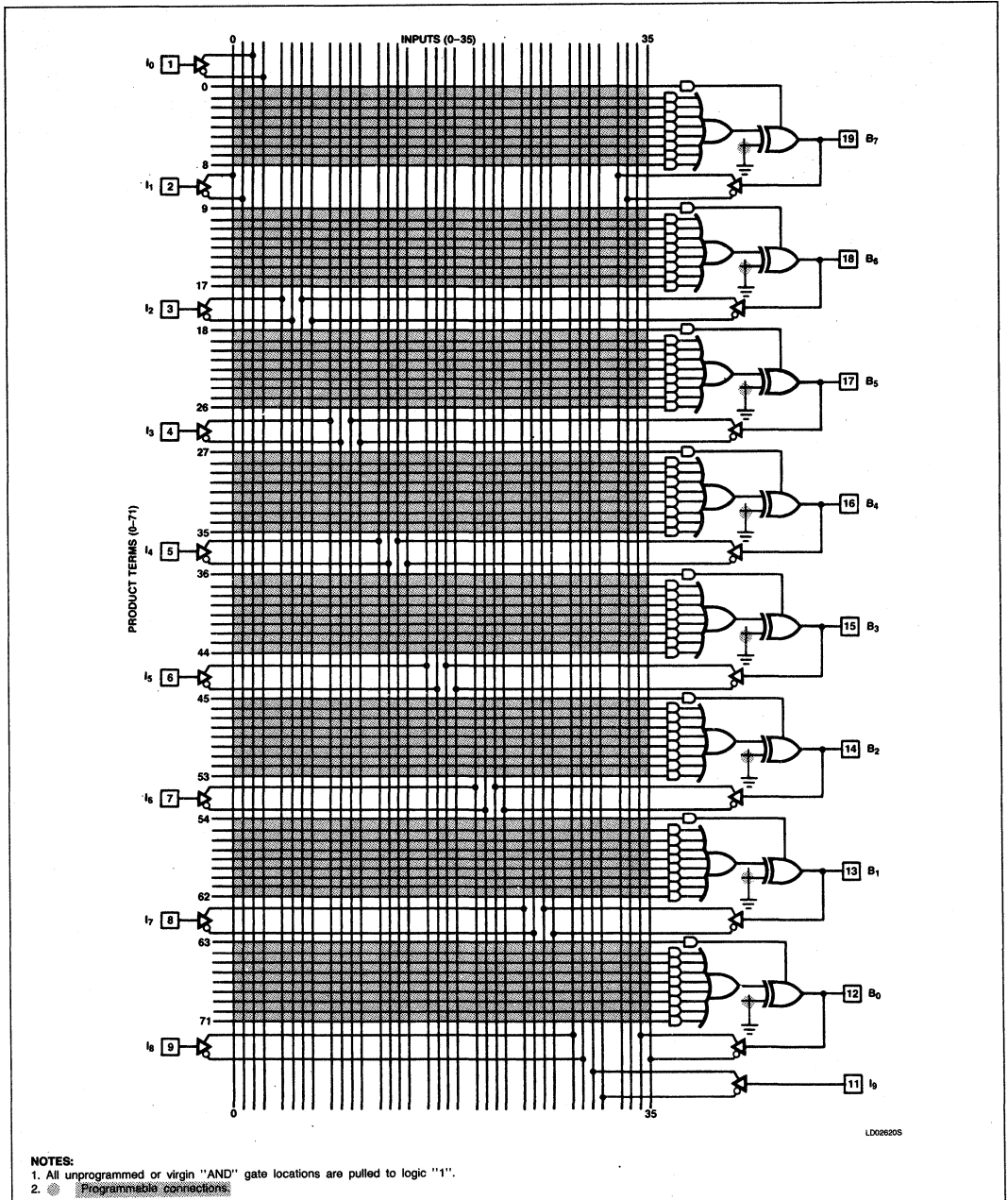
- 100% functional replacement for all 20-pin combinatorial PALs\*
- Random logic
- Code converters
- Fault detectors
- Function generators
- Address mapping
- Multiplexing

\*PAL is a registered trademark of Monolithic Memories, Inc.

# Programmable AND Array Logic (18 × 72 × 8)

PLHS18P8A

## FPLA LOGIC DIAGRAM



## Programmable AND Array Logic (18 × 72 × 8)

PLHS18P8A

## ORDERING INFORMATION

DESCRIPTION	ORDER CODE
20-pin Plastic DIP 300mil-wide	PLHS18P8AN
20-pin Plastic Leaded Chip Carrier	PLHS18P8AA

## THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS<sup>1</sup>

SYMBOL	PARAMETER	RATING		UNIT
		MIN	MAX	
V <sub>CC</sub>	Supply voltage	-0.5	+7	V <sub>DC</sub>
V <sub>IN</sub>	Input voltage	-0.5	+5.5	V <sub>DC</sub>
V <sub>OUT</sub>	Output voltage	-0.5	V <sub>CC</sub> Max	V <sub>DC</sub>
V <sub>OUTPRG</sub>	Output voltage (programming)		+21	V <sub>DC</sub>
I <sub>IN</sub>	Input current	-30	+5	mA
I <sub>OUT</sub>	Output current		+100	mA
I <sub>OUTPRG</sub>	Output current (programming)		+170	mA
T <sub>A</sub> T <sub>STG</sub>	Temperature range Operating temp Storage temp	0 -65	+75 +150	°C °C

Notes on following page.

4

## Programmable AND Array Logic (18 × 72 × 8)

PLHS18P8A

DC ELECTRICAL CHARACTERISTICS  $0^{\circ}\text{C} \leq T_A \leq 75^{\circ}\text{C}$ ,  $4.75\text{V} \leq V_{CC} \leq 5.25\text{V}$ 

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ <sup>2</sup>	Max	
<b>Input voltage<sup>3</sup></b>						
$V_{IL}$ $V_{IH}$ $V_I$	Low High Clamp	$V_{CC} = \text{Min}$ $V_{CC} = \text{Max}$ $V_{CC} = \text{Min}, I_{IN} = -18\text{mA}$	+2.0	-0.9	+0.8 -1.2	V
<b>Output voltage</b>						
$V_{OL}$ $V_{OH}$	Low High	$V_{CC} = \text{Min}, V_{IN} = V_{IH} \text{ or } V_{IL}$ $I_{OL} = +24\text{mA}$ $I_{OH} = -3.2$	+2.4	+3.5	+0.50	V
<b>Input current</b>						
$I_{IL}$ $I_{IH}$ $I_I$	Low High High	$V_{CC} = \text{Max}$ $V_{IN} = +0.40\text{V}$ $V_{IN} = +2.7\text{V}$ $V_{IN} = +5.5\text{V}$		-20	-100 +25 +1.0	$\mu\text{A}$ $\mu\text{A}$ mA
<b>Output current</b>						
$I_{OZH}$ $I_{OZL}$ $I_{SC}$	Output leakage Output leakage Short circuit <sup>4</sup>	$V_{CC} = \text{Max}, V_{IL} = 0.8\text{V}, V_{IH} = 2.0\text{V}$ $V_{OUT} = +2.7\text{V}$ $V_{OUT} = +0.40\text{V}$ $V_{OUT} = +0.5\text{V}$	-25	-60	+100 -250 -90	$\mu\text{A}$ $\mu\text{A}$ mA
$I_{CC}$	$V_{CC}$ current	$V_{CC} = \text{Max}, \text{All inputs} = \text{GND}$		100	155	mA
<b>Capacitance<sup>5</sup></b>						
$C_{IN}$ $C_{OUT}$	Input I/O	$V_{CC} = +5\text{V}$ $V_{IN} = 2.0\text{V} @ f = 1\text{MHz}$ $V_{OUT} = 2.0\text{V} @ f = 1\text{MHz}$		6 9		pF pF

**NOTES:**

- Stresses above those listed under "Absolute Maximum Ratings" may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.
- Typical limits are at  $V_{CC} = 5.0\text{V}$  and  $T_A = +25^{\circ}\text{C}$ .
- These are absolute values with respect to device ground and all overshoots due to system or tester noise are included.
- Not more than one output should be tested at a time. Duration of the short circuit should not be more than one second.  $V_{OUT} = 0.5\text{V}$  has been chosen to avoid test problems caused by tester ground degradation.
- These parameters are not 100% tested, but are periodically sampled.

# Programmable AND Array Logic (18 × 72 × 8)

PLHS18P8A

## AC ELECTRICAL CHARACTERISTICS $R_1 = 200\Omega, R_2 = 390\Omega, 0^\circ\text{C} \leq T_A \leq +75^\circ\text{C}, 4.75\text{V} \leq V_{CC} \leq 5.25\text{V}$

SYMBOL	PARAMETER	TO	FROM	TEST CONDITIONS	LIMITS			UNIT
					Min	Typ	Max	
$t_{PD}$	Propagation delay	Input $\pm$	Output $\pm$	$C_L = 50\text{pF}$		14	20	ns
$t_{EA}$	Output enable	Input $\pm$	Output -	$C_L = 50\text{pF}$		14	20	ns
$t_{ER}$	Output disable	Input $\pm$	Output +	$C_L = 5\text{pF}$		14	20	ns

**NOTES:**

1. Typical limits are at  $V_{CC} = 5.0\text{V}$  and  $T_A = +25^\circ\text{C}$ .
2.  $t_{PD}$  is tested with switch  $S_1$  closed and  $C_L = 50\text{pF}$ .
3. For Tri-state output; output enable times are tested with  $C_L = 50\text{pF}$  to the 1.5V level, and  $S_1$  is open for high-impedance to High tests and closed for high-impedance to Low tests. Output disable times are tested with  $C_L = 5\text{pF}$ . High-to-High impedance tests are made to an output voltage of  $V_{OH} = -0.5\text{V}$  with  $S_1$  open, and Low-to-High impedance tests are made to the  $V_{OL} = +0.5\text{V}$  level with  $S_1$  closed.

**VIRGIN STATE**

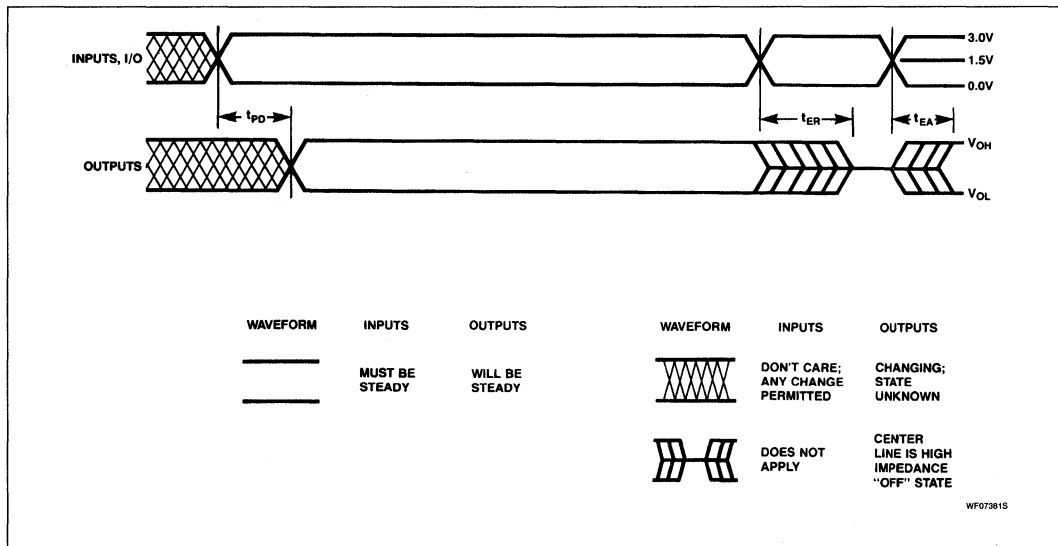
A factory shipped virgin device contains all fusible links open, such that:

1. All outputs are at "L" polarity.
2. All outputs are enabled.
3. All p-terms are enabled.

**TIMING DEFINITIONS**

- $t_{PD}$  Input to output propagation delay.
- $t_{ER}$  Input to output disable (Tri-state) delay (Output Disable).
- $t_{EA}$  Input to Output Enable delay (Output Enable).

**TIMING DIAGRAM**



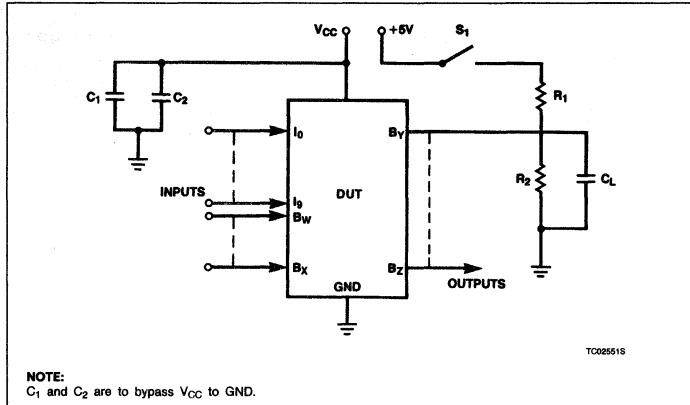
4



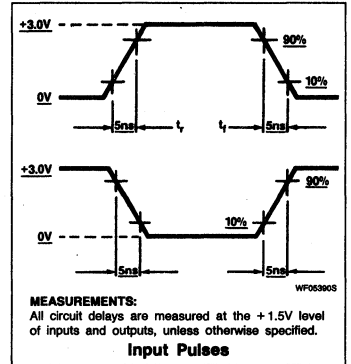
# Programmable AND Array Logic (18 × 72 × 8)

# PLHS18P8A

## AC TEST LOAD CIRCUIT



## VOLTAGE WAVEFORM



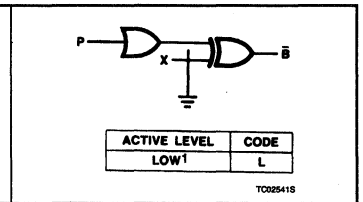
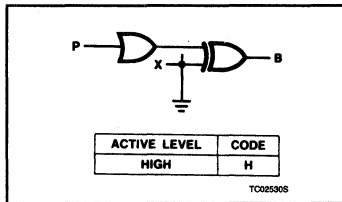
## LOGIC PROGRAMMING

The FPLA can be programmed by means of Logic programming equipment.

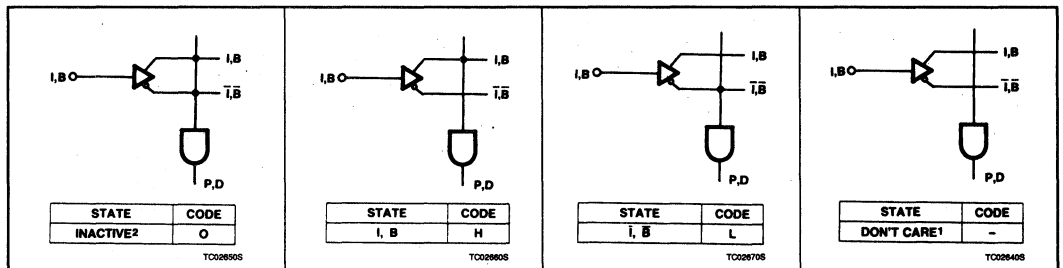
With Logic programming, the AND/EX-OR gate input connections necessary to implement the desired logic function are coded directly from logic equations using the Program Table on the following page.

In this Table the logic state of variables I, P, and B associated with each Sum Term, S, is assigned a symbol which results in the proper fusing pattern of corresponding link pairs, defined as follows:

## OUTPUT POLARITY - (B)



## "AND" ARRAY - (I, B)



### NOTES:

- This is the initial state of all link pairs.
- All unused product terms must be programmed with all pairs of fuses in the INACTIVE state (all fuses on an unused p-term must be programmed).





### INDEX

#### Section 5 - PLD Data Sheets

##### Series 24

PLS161	Field-Programmable Logic Array (12 × 48 × 8) .....	5-3
PLS162	Field-Programmable Address Decoder (16 × 5) .....	5-9
PLS163	Field-Programmable Address Decoder (12 × 9) .....	5-15
PLS167	Field-Programmable Logic Sequencer (14 × 48 × 6) ...	5-21
PLS167A	Field-Programmable Logic Sequencer; Enhanced Speed (14 × 48 × 6) .....	5-33
PLS168	Field-Programmable Logic Sequencer (12 × 48 × 8) ...	5-45
PLS168A	Field-Programmable Logic Sequencer; Enhanced Speed (12 × 48 × 8) .....	5-57
PLS173	Field-Programmable Logic Array (22 × 42 × 10) .....	5-69
PLS179	Field-Programmable Logic Sequencer (20 × 45 × 12) ..	5-75
PLHS473	Field-Programmable Logic Array (20 × 24 × 11) .....	5-85
PLC473	Erasable-Programmable Logic Array (20 × 24 × 11)....	5-91



## PLS161 Field-Programmable Logic Array (12 × 48 × 8)

Signetics Programmable Logic  
Product Specification

### Application Specific Products ● Series 24

#### DESCRIPTION

The PLS161 is a bipolar, Field-Programmable Logic Array (FPLA). The device utilizes the standard AND/OR/Invert architecture to directly implement custom sum of product logic equations.

This device consists of 12 dedicated inputs and 8 dedicated outputs. Each output is capable of being actively controlled by any or all of the 48 product terms. The True, Complement, or Don't Care condition of each of the 12 inputs ANDed together comprise one P-term. All 48 P-terms are selectively ORed to each output. The user must then only select which P-terms will activate an output by disconnecting terms which do not affect the output. In addition each output can be fused as Active-High (H) or Active-Low (L).

The PLS161 is fully TTL compatible, and includes a chip enable input for output inhibit control and expansion of input variables.

Order codes are contained on the pages following.

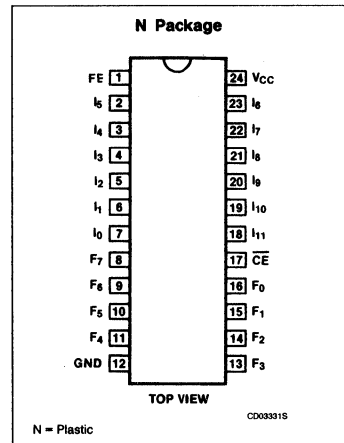
#### FEATURES

- Field-Programmable (Ni-Cr Link)
- Input variables: 12
- Output functions: 8
- Product terms: 48
- I/O propagation delay: 50ns (max.)
- Power dissipation: 600mW (typ.)
- Input loading: -100µA (max.)
- Chip Enable Input
- Output disable function
- Separate I/O architecture
- Tri-state outputs

#### APPLICATIONS

- CRT display systems
- Random logic
- Code conversion
- Peripheral controllers
- Function generators
- Look-up and decision tables
- Microprogramming
- Address mapping
- Character generators
- Sequential controllers
- Data security encoders
- Fault detectors
- Frequency synthesizers

#### PIN CONFIGURATION



#### LOGIC FUNCTION

##### TYPICAL PRODUCT TERM

$$P_0 = I_0 \cdot I_1 \cdot I_2 \cdot I_3 \cdot I_{11}$$

##### TYPICAL OUTPUT FUNCTIONS:

$$\text{@ CE} = 0:$$

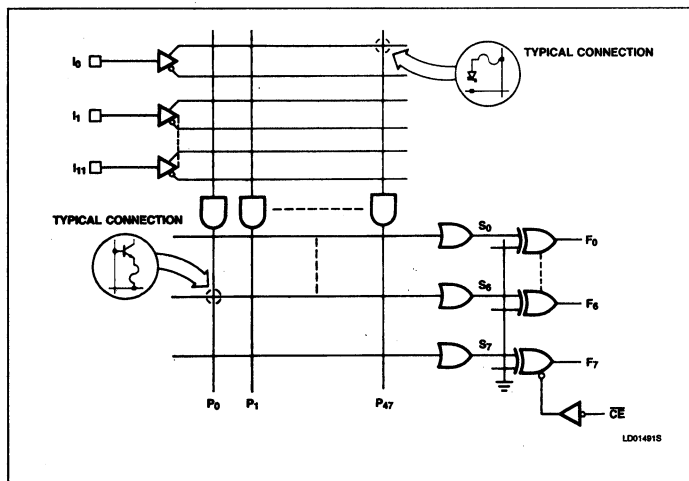
$$F_0 = (P_0 + P_1 + P_2) \text{ @ L = CLOSED}$$

$$F_0 = (P_0 \cdot P_1 \cdot P_2) \text{ @ L = OPEN}$$

##### NOTE:

For each of the 8 outputs either the function  $F_p$  (Active-High) or  $\bar{F}_p$  (Active-Low) is available, but not both. The desired output polarity is programmed via the EX-OR gates.

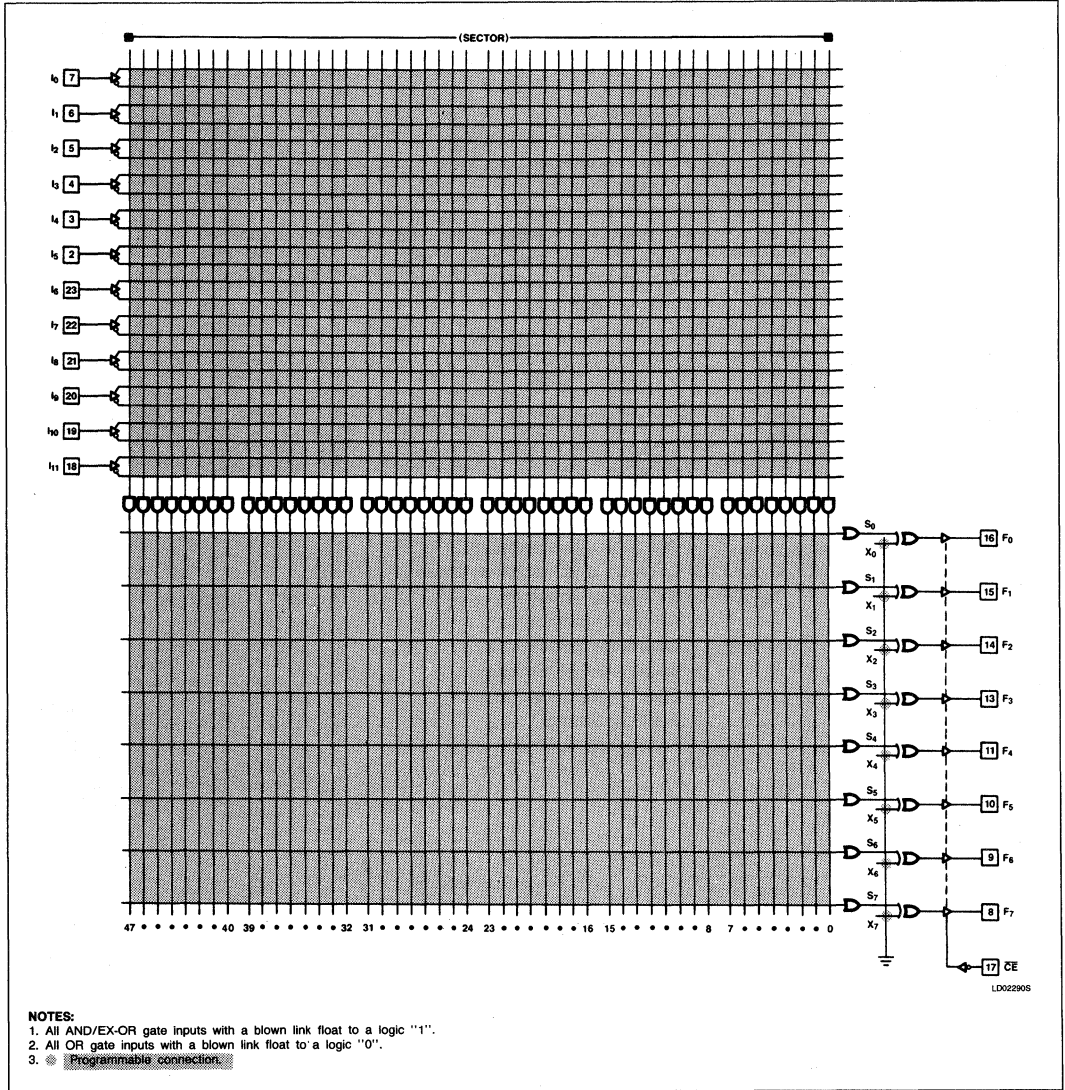
#### FUNCTIONAL DIAGRAM



# Field-Programmable Logic Array (12 × 48 × 8)

PLS161

## FPLA LOGIC DIAGRAM



## Field-Programmable Logic Array (12 × 48 × 8)

PLS161

## ORDERING INFORMATION

DESCRIPTION	ORDER CODE
24-pin Plastic DIP 300mil-wide	PLS161N

## THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS<sup>1</sup>

SYMBOL	PARAMETER	RATING		UNIT
		Min	Max	
V <sub>CC</sub>	Supply voltage		+7	V <sub>DC</sub>
V <sub>IN</sub>	Input voltage		+5.5	V <sub>DC</sub>
V <sub>OUT</sub>	Output voltage		+5.5	V <sub>DC</sub>
I <sub>IN</sub>	Input currents	-30	+30	mA
I <sub>OUT</sub>	Output currents		+100	mA
T <sub>A</sub> T <sub>STG</sub>	Temperature range Operating Storage	0 -65	+75 +150	°C

DC ELECTRICAL CHARACTERISTICS 0°C ≤ T<sub>A</sub> ≤ 75°C, 4.75V ≤ V<sub>CC</sub> ≤ 5.25V

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ <sup>2</sup>	Max	
<b>Input voltage<sup>3</sup></b>						
V <sub>IH</sub> V <sub>IL</sub> V <sub>IC</sub>	High Low Clamp <sup>4</sup>	V <sub>CC</sub> = Max V <sub>CC</sub> = Min V <sub>CC</sub> = Min, I <sub>IN</sub> = -12mA	2		0.8 -1.2	V
<b>Output voltage<sup>3</sup></b>						
V <sub>OH</sub> V <sub>OL</sub>	High <sup>5</sup> Low <sup>6</sup>	V <sub>CC</sub> = Min I <sub>OH</sub> = -2mA I <sub>OL</sub> = 9.6mA	2.4	0.35	0.45	V
<b>Input current</b>						
I <sub>IH</sub> I <sub>IL</sub>	High Low	V <sub>CC</sub> = Max V <sub>IN</sub> = 5.5V V <sub>IN</sub> = 0.45V		< 1 -10	40 -100	μA
<b>Output current</b>						
I <sub>O(OFF)</sub> I <sub>OS</sub>	Hi-Z state Short circuit <sup>4,7</sup>	CE = High, V <sub>CC</sub> = Max V <sub>OUT</sub> = 5.5V V <sub>OUT</sub> = 0.45V CE = Low, V <sub>OUT</sub> = 0V		1 -1	40 -40 -70	μA mA
I <sub>CC</sub>	V <sub>CC</sub> supply current <sup>8</sup>	V <sub>CC</sub> = Max		120	170	mA
<b>Capacitance</b>						
C <sub>IN</sub> C <sub>OUT</sub>	Input Output	CE = High, V <sub>CC</sub> = 5.0V V <sub>IN</sub> = 2V V <sub>OUT</sub> = 2V		8 17		pF

Notes on following page.



# Field-Programmable Logic Array (12 × 48 × 8)

PLS161

## AC ELECTRICAL CHARACTERISTICS $R_1 = 470\Omega$ , $R_2 = 1k\Omega$ , $C_L = 30pF$ , $0^\circ C < T_A \leq +75^\circ C$ , $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TO	FROM	LIMITS			UNIT
				Min	Typ <sup>2</sup>	Max	
<b>Propagation delay</b>							
$t_{PD}$	Input	Output	Input		35	50	ns
$t_{CE}$	Chip enable	Output	Chip enable		15	30	ns
<b>Disable time</b>							
$t_{CD}$	Chip disable	Output	Chip enable		15	30	ns

**NOTES:**

1. Stresses above those listed under "Absolute Maximum Ratings" may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other conditions above those indicated in the operational and programming specification of the device is not implied.
2. All values are at  $V_{CC} = 5V$ ,  $T_A = +25^\circ C$ .
3. All voltage values are with respect to network ground terminal.
4. Test one pin at a time.
5. Measured with  $V_{IL}$  applied to  $\overline{CE}$  and a logic high stored.
6. Measured with a programmed logic condition for which the output test is at a low logic level. Output sink current is applied through a resistor to  $V_{CC}$ .
7. Duration of short circuit should not exceed 1 second.
8.  $I_{CC}$  is measured with the chip enable input grounded, all other inputs at 4.5V and the outputs open.

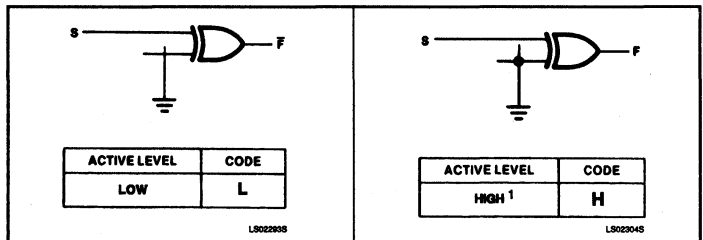
### LOGIC PROGRAMMING

The FPLA can be programmed by means of Logic programming equipment.

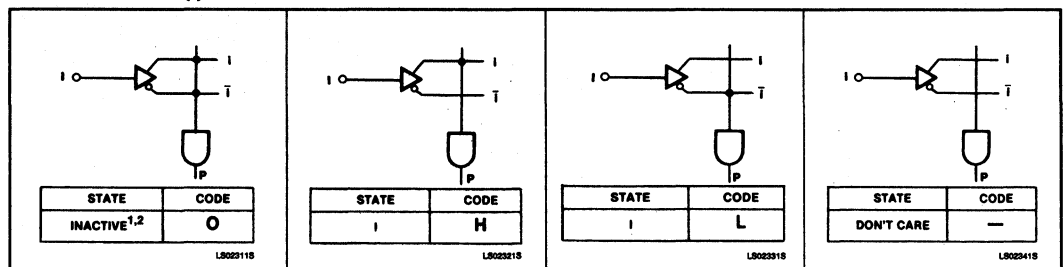
With Logic programming, the AND/OR/EX-OR gate input connections necessary to implement the desired logic function are coded directly from logic equations using the Program Table on the following page.

In this Table the logic state or action of variables I, P, and F, associated with each Sum Term  $S_n$ , is assigned a symbol which results in the proper fusing pattern of corresponding link pairs, defined as follows:

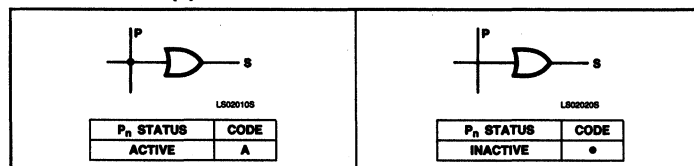
### EX-OR ARRAY - (F)



### "AND" ARRAY - (I)



### "OR" ARRAY - (F)



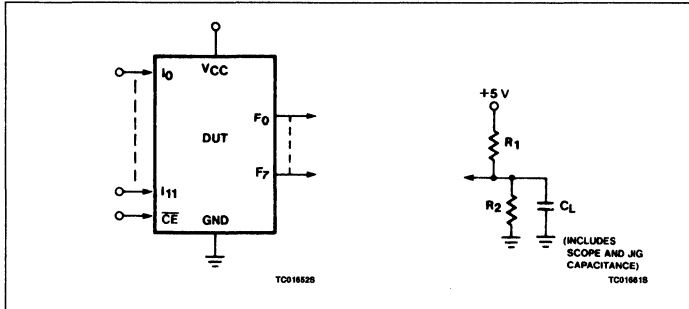
**NOTES:**

1. This is the initial unprogrammed state of all link pairs. It is normally associated with all unused (inactive) AND gates  $P_n$ .
2. Any gate  $P_n$  will be unconditionally inhibited if any one of its (I) link pairs is left intact.

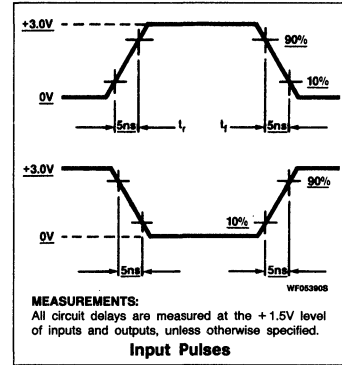
# Field-Programmable Logic Array (12 × 48 × 8)

PLS161

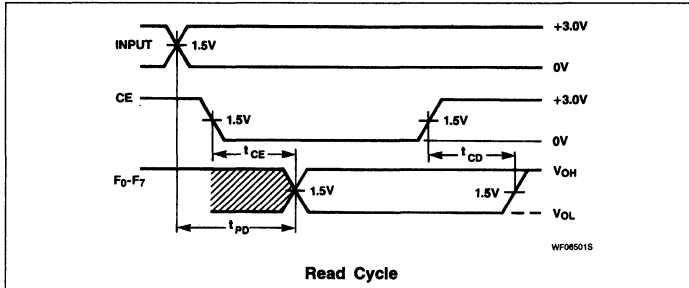
## TEST LOAD CIRCUIT



## VOLTAGE WAVEFORM



## TIMING DIAGRAMS



## TIMING DEFINITIONS

- $t_{CE}$  Delay between beginning of Chip Enable Low (with Input valid) and when Data Output becomes valid.
- $t_{CD}$  Delay between when Chip Enable becomes High and Data Output is in off state (Hi-Z or High).
- $t_{PD}$  Delay between beginning of valid Input (with Chip Enable Low) and when Data Output becomes valid.

## VIRGIN STATE

The PLS161 virgin device is factory shipped in an unprogrammed state, with all fuses intact, such that:

1. All  $P_n$  terms are disabled. (inactive).
2. All  $P_n$  terms are active on all outputs.
3. All outputs are Active-High.



## PLS162 Field-Programmable Address Decoder (16 × 5)

Signetics Programmable Logic  
Product Specification

### Application Specific Products • Series 24

#### DESCRIPTION

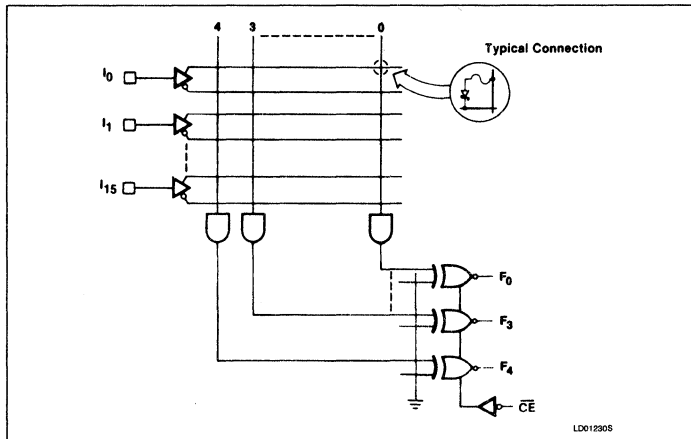
The PLS162 is a bipolar, Field-Programmable Address Decoder. The device consists of five AND/NAND gates which share 16 common inputs. The type of gate is selected by programming the output as Active-High (H) or Active-Low (L). Each of the 16 inputs  $I_0 - I_{15}$  can be programmed to provide the True (H), Complement (L), or Don't Care (—) state to each of the five AND/NAND gates. OR/NOR logic functions can also be implemented by complementing the inputs and outputs via on-chip inverting buffers.

The device is field programmable, which means that custom patterns are immediately available.

The PLS162 includes chip-enable control for output strobing and inhibit. It features Tri-state outputs for ease of expansion of input variables and application in bus-organized systems.

Order codes are contained on the following pages.

#### FUNCTIONAL DIAGRAM



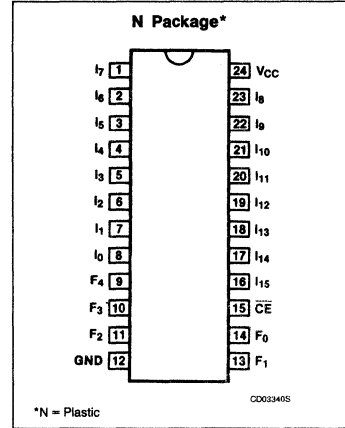
#### FEATURES

- Field-Programmable (Ni-Cr link)
- 16 input variables
- 5 output functions
- Chip Enable Input
- I/O propagation delay: 30ns (max.)
- Power dissipation: 500mW (typ.)
- Input loading:  $-100\mu\text{A}$  (max.)
- Tri-state outputs
- Output disable function: Hi-Z
- Fully TTL compatible

#### APPLICATIONS

- Random logic
- Address decoders
- Code detectors
- Peripheral selectors
- Fault monitors
- Machine state decoders

#### PIN CONFIGURATION



#### LOGIC FUNCTION

##### TYPICAL OUTPUT FUNCTIONS:

##### ACTIVE-HIGH

$$X = A \cdot \bar{B} \cdot C \cdot \dots$$

##### ACTIVE-LOW

$$X = A \cdot \bar{B} \cdot C \cdot \dots$$

$$X = \bar{A} + B + C + \dots$$

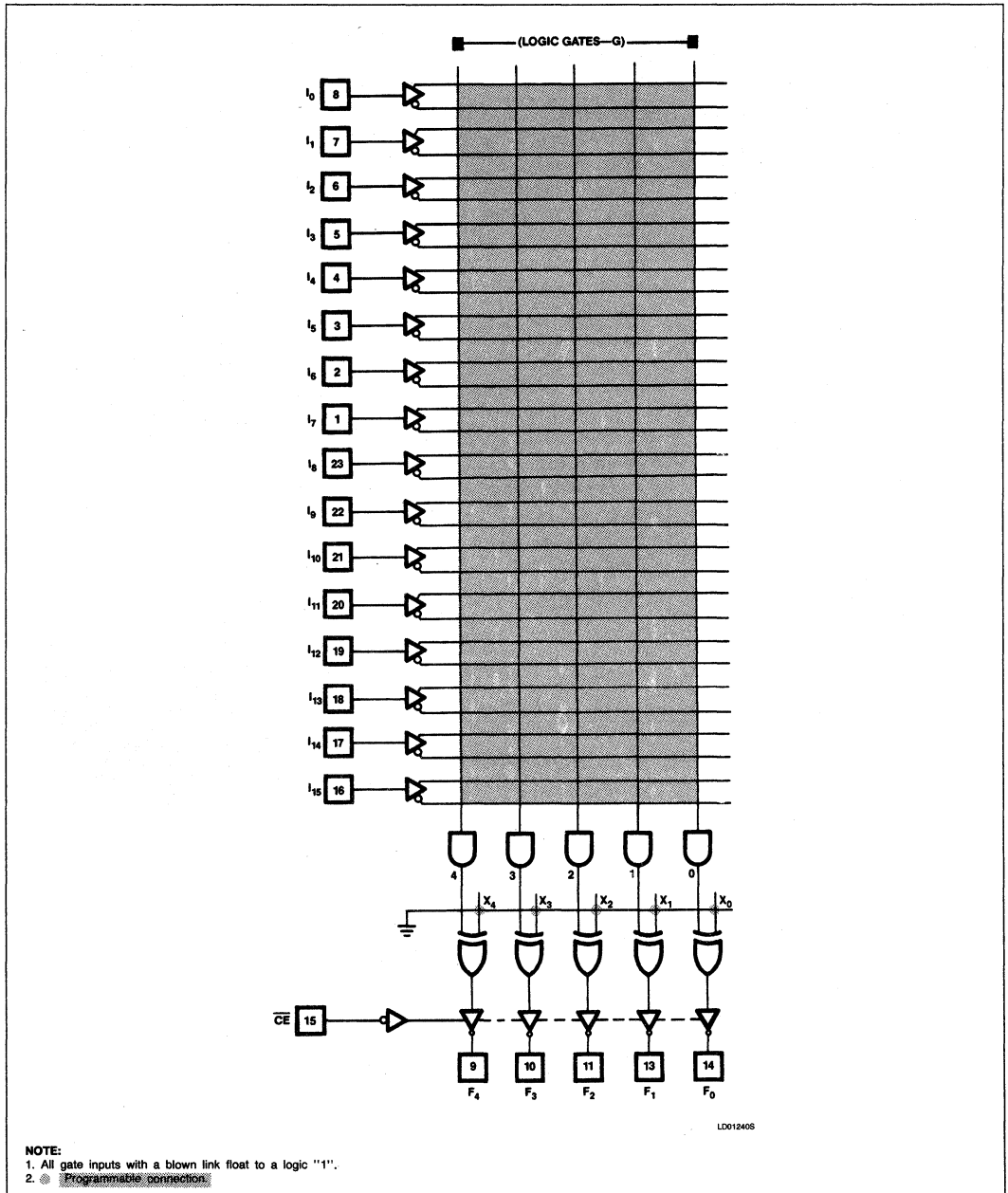
##### NOTES:

1. For each of the 5 outputs, either function X (Active-High) or  $\bar{X}$  (Active-Low) is available, but not both. The desired output polarity is programmed via the EX-OR gates.
2. X, A, B, C, etc. are user defined connections to fixed inputs (I) and output pins (F).

# Field-Programmable Address Decoder (16 × 5)

PLS162

## FPAD LOGIC DIAGRAM



## Field-Programmable Address Decoder (16 × 5)

PLS162

## ORDERING INFORMATION

DESCRIPTION	ORDER CODE
24-pin Plastic DIP 300mil-wide	PLS162N

## ABSOLUTE MAXIMUM RATINGS

SYMBOL	PARAMETER	RATING	UNIT
V <sub>CC</sub>	Supply voltage	+7	V <sub>DC</sub>
V <sub>IN</sub>	Input voltage	+5.5	V <sub>DC</sub>
V <sub>O</sub>	Output voltage Off-state	+5.5	V <sub>DC</sub>
I <sub>IN</sub>	Input current	±30	mA
I <sub>OUT</sub>	Output current	+100	mA
T <sub>A</sub> T <sub>STG</sub>	Temperature range Operating Storage	0 to +75 -65 to +150	°C

DC ELECTRICAL CHARACTERISTICS 0°C ≤ T<sub>A</sub> ≤ +75°C, 4.75 ≤ V<sub>CC</sub> ≤ 5.25V

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ <sup>2</sup>	Max	
<b>Input voltage<sup>1</sup></b>						
V <sub>IH</sub> V <sub>IL</sub> V <sub>IC</sub>	High Low Clamp <sup>3</sup>	V <sub>CC</sub> = Max V <sub>CC</sub> = Min V <sub>CC</sub> = Min, I <sub>IN</sub> = -12mA	2.0		0.8 -1.2	V
<b>Output voltage<sup>1</sup></b>						
V <sub>OH</sub> V <sub>OL</sub>	High <sup>5</sup> Low <sup>4</sup>	V <sub>CC</sub> = Min I <sub>OH</sub> = -2mA I <sub>OL</sub> = 15mA	2.4	0.35	0.5	V
<b>Input current</b>						
I <sub>IH</sub> I <sub>IL</sub>	High Low	V <sub>CC</sub> = Max V <sub>IN</sub> = 5.5V V <sub>IN</sub> = 0.45V		< 1 -10	40 -100	μA
<b>Output current</b>						
I <sub>O</sub> (OFF) I <sub>OS</sub>	Hi-Z state <sup>6</sup> Short circuit <sup>3,7</sup>	V <sub>CC</sub> = Max V <sub>OUT</sub> = 5.5V V <sub>OUT</sub> = 0.45V V <sub>OUT</sub> = 0V		1 -1	40 -40 -70	μA mA
I <sub>CC</sub>	V <sub>CC</sub> supply current <sup>8</sup>	V <sub>CC</sub> = Max		100	155	mA
<b>Capacitance</b>						
C <sub>IN</sub> C <sub>OUT</sub>	Input Output <sup>6</sup>	V <sub>CC</sub> = 5.0V V <sub>IN</sub> = 2.0V V <sub>OUT</sub> = 2.0V		8 15		pF

## Field-Programmable Address Decoder (16 × 5)

PLS162

**AC ELECTRICAL CHARACTERISTICS**  $0^{\circ}\text{C} \leq T_A \leq +75^{\circ}\text{C}$ ,  $4.75 \leq V_{CC} \leq 5.25\text{V}$ ,  $R_1 = 470\Omega$ ,  $R_2 = 1\text{k}\Omega$ 

SYMBOL	PARAMETER	TO	FROM	TEST CONDITIONS	LIMITS			UNIT
					Min	Typ <sup>1</sup>	Max	
$t_{PD}$	Propagation delay	Output	Input	$C_L = 30\text{pF}$		20	30	ns
$t_{CE}$	Chip enable	Output	Chip enable	$C_L = 30\text{pF}$		20	30	ns
$t_{CD}$	Chip disable	Output	Chip enable	$C_L = 5\text{pF}$		20	30	ns

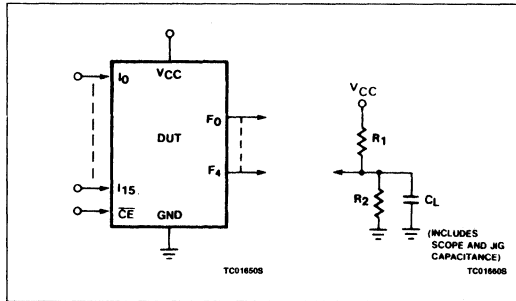
**NOTES:**

1. All voltage values are with respect to network ground terminal.
2. All typical values are at  $V_{CC} = 5\text{V}$ ,  $T_A = +25^{\circ}\text{C}$ .
3. Test each pin one at a time.
4. Measure with a programmed logic condition for which the output under test is at low logic level. Output sink current is supplied through a resistor to  $V_{CC}$ .
5. Measured with  $V_{IL}$  applied to  $\overline{CE}$  and logic high at the output.
6. Measured with  $V_{IH}$  applied to  $\overline{CE}$ .
7. Duration of short circuit should not exceed 1 second.
8.  $I_{CC}$  is measured with the outputs open.

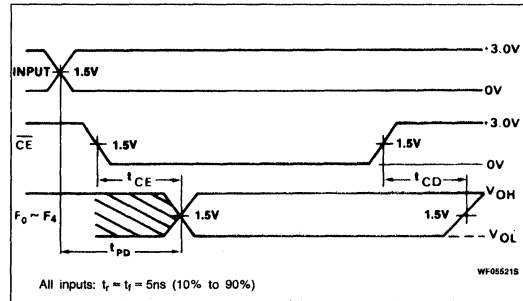
# Field-Programmable Address Decoder (16 × 5)

# PLS162

### TEST LOAD CIRCUIT



### VOLTAGE WAVEFORM



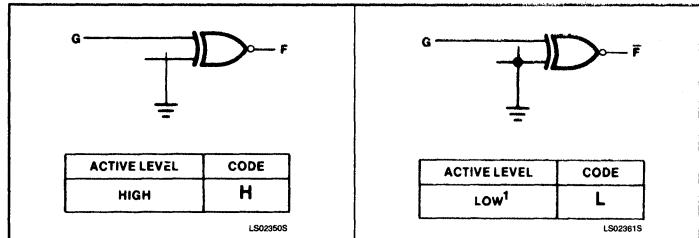
### LOGIC PROGRAMMING

In a virgin device all Ni-Cr links are intact. The FPAD can be programmed by means of Logic Programming equipment.

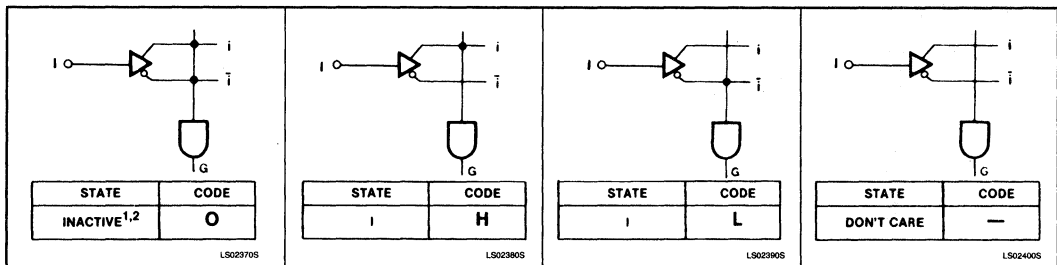
With Logic programming, the AND/EX-OR gate input connections necessary to implement the desired logic function are coded directly from the logic equations using the Program Table on the following page.

In this table, the logic state of variables I and F associated with each gate  $G_n$  is assigned a symbol which results in the proper fusing pattern of corresponding links defined as follows:

### OUTPUT POLARITY - (F)



### "AND" ARRAY - (I)



#### NOTES:

1. This is the initial unprogrammed state of all links.
2. Any gate  $G_n$  will be unconditionally inhibited if both the true and complement of any input (either I or B) are left intact.

### VIRGIN DEVICE

The PLS162 is shipped in an unprogrammed state, characterized by:

1. All  $P_n$  terms are disabled. (Inactive).
2. All  $P_n$  terms are active on all outputs.
3. All outputs are Active-Low.



# Field-Programmable Address Decoder (16 × 5)

PLS162

## FPAD PROGRAM TABLE

CUSTOMER NAME _____ PURCHASE ORDER # _____ SIGNETICS DEVICE # _____ TOTAL NUMBER OF PARTS _____ PROGRAM TABLE # _____	THIS PORTION TO BE COMPLETED BY SIGNETICS CF (XXXX) _____ CUSTOMER SYMBOLIZED PART # _____ DATE RECEIVED _____ COMMENTS _____
---	---

F<sub>0</sub> (14) \_\_\_\_\_ = \_\_\_\_\_  
 F<sub>1</sub> (13) \_\_\_\_\_ = \_\_\_\_\_  
 F<sub>2</sub> (11) \_\_\_\_\_ = \_\_\_\_\_  
 F<sub>3</sub> (10) \_\_\_\_\_ = \_\_\_\_\_  
 F<sub>4</sub> (9) \_\_\_\_\_ = \_\_\_\_\_

GATE		INPUT															
POLARITY		I <sub>15</sub>	I <sub>14</sub>	I <sub>13</sub>	I <sub>12</sub>	I <sub>11</sub>	I <sub>10</sub>	I <sub>9</sub>	I <sub>8</sub>	I <sub>7</sub>	I <sub>6</sub>	I <sub>5</sub>	I <sub>4</sub>	I <sub>3</sub>	I <sub>2</sub>	I <sub>1</sub>	I <sub>0</sub>
F <sub>0</sub>																	
F <sub>1</sub>																	
F <sub>2</sub>																	
F <sub>3</sub>																	
F <sub>4</sub>																	
PIN NO.		16	17	18	19	20	21	22	23	1	2	3	4	5	6	7	8
VARIABLE NAME																	

NOTES 1. The FPAD is shipped with all links intact. Thus a background of entries corresponding to states of virgin links exists in the table, shown BLANK for clarity. 2. Unused I bits are normally programmed Don't Care (—). 3. Unused Gates can be left blank.	PROGRAM TABLE ENTRIES	AND <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr><td>INACTIVE</td><td>0</td></tr> <tr><td>I</td><td>H</td></tr> <tr><td>I</td><td>L</td></tr> <tr><td>Don't Care</td><td>—</td></tr> </table> (I)	INACTIVE	0	I	H	I	L	Don't Care	—	CONTROL <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr><td>HIGH</td><td>H</td></tr> <tr><td>LOW</td><td>L</td></tr> </table> (POL)	HIGH	H	LOW	L
INACTIVE	0														
I	H														
I	L														
Don't Care	—														
HIGH	H														
LOW	L														

TB012815

# PLS163

## Field-Programmable Address Decoder (12 × 9)

Signetics Programmable Logic  
Product Specification

Application Specific Products  
• Series 24

### DESCRIPTION

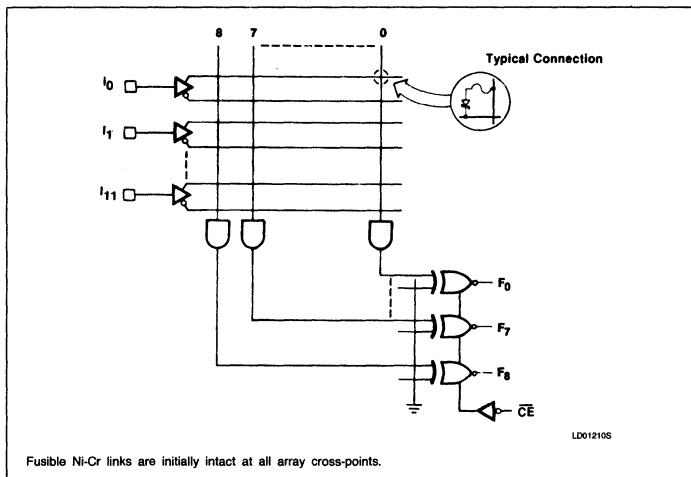
The PLS163 is a bipolar, Field-Programmable Address Decoder. The device consists of nine AND/NAND gates which share 12 common inputs. The type of gate is selected by programming the output as Active-High (H) or Active-Low (L). Each of the 12 inputs  $I_0 - I_{11}$  can be programmed to provide the True (H), Complement (L), or Don't Care (—) state to each of the nine AND/NAND gates. OR/NOR logic functions can also be implemented by complementing the inputs and outputs via on-chip inverting buffers.

The device is field programmable, which means that custom patterns are immediately available.

The PLS163 includes chip-enable control for output strobing and inhibit. It features Tri-state outputs for ease of expansion of input variables and application in bus-organized systems.

Order codes are contained on the following pages.

### FUNCTIONAL DIAGRAM



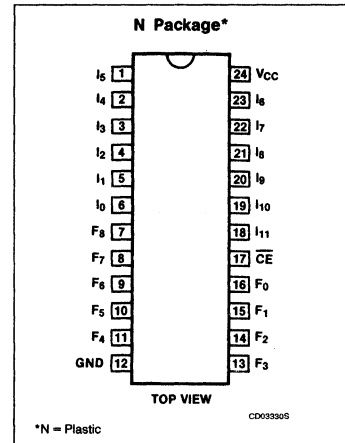
### FEATURES

- Field-Programmable (Ni-Cr link)
- 12 input variables
- 9 output functions
- Chip Enable input
- I/O propagation delay: 30ns (max.)
- Power dissipation: 600mW (typ.)
- Input loading:  $-100\mu\text{A}$  (max.)
- Tri-state outputs
- Output disable function: HI-Z
- Fully TTL compatible

### APPLICATIONS

- Random logic
- Address decoders
- Code detectors
- Peripheral selectors
- Fault monitors
- Machine state decoders

### PIN CONFIGURATION



### LOGIC FUNCTION

#### TYPICAL OUTPUT FUNCTIONS:

##### ACTIVE-HIGH

$$X = A \cdot \bar{B} \cdot C \cdot \dots$$

##### ACTIVE-LOW

$$X = A \cdot \bar{B} \cdot C \cdot \dots$$

$$X = \bar{A} + B + C + \dots$$

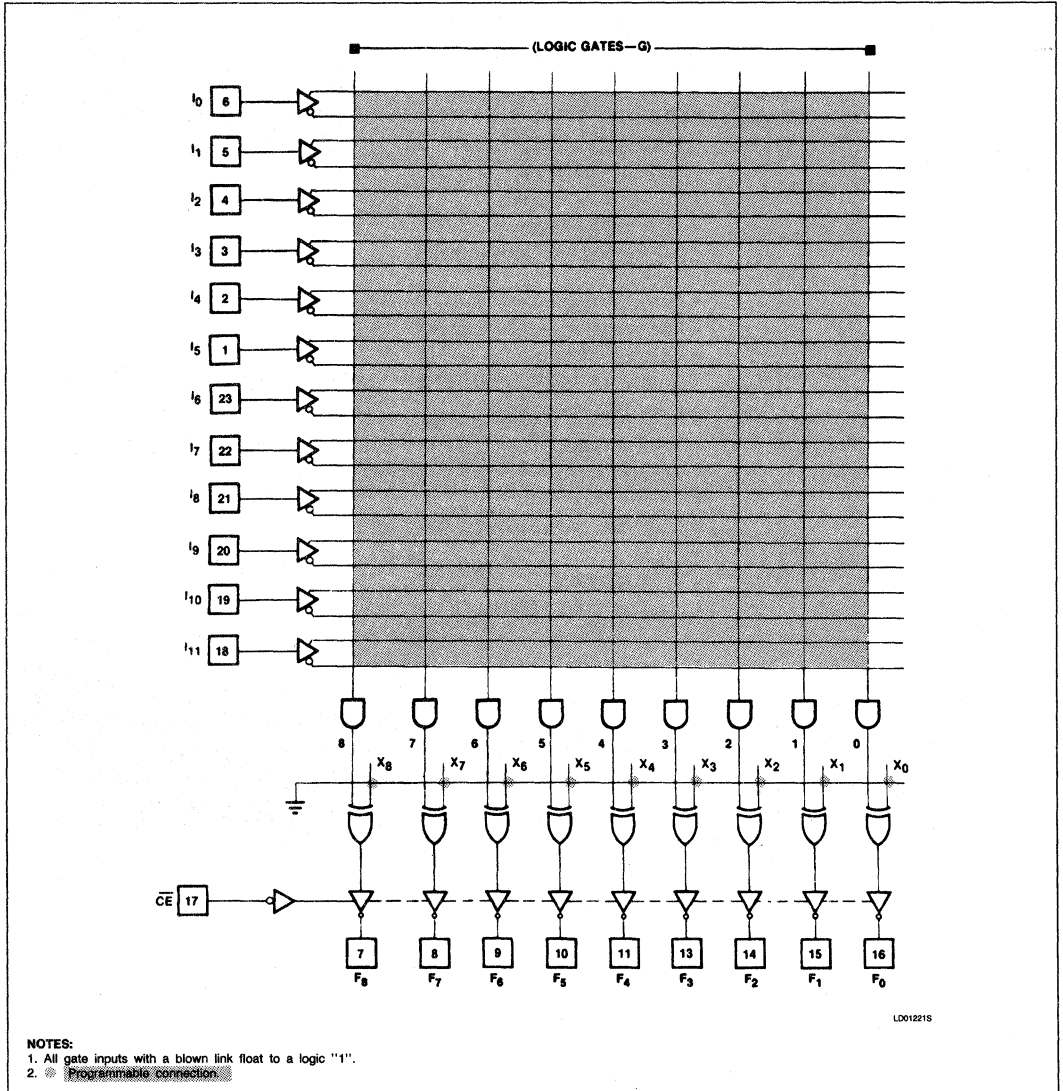
#### NOTES:

1. For each of the 9 outputs, either function X (Active-High) or  $\bar{X}$  (Active-Low) is available, but not both. The desired output polarity is programmed via the EX-OR gates.
2. X, A, B, C, etc. are user defined connections to fixed inputs (I) and output pins (F).

# Field-Programmable Address Decoder (12 × 9)

PLS163

## FPAD LOGIC DIAGRAM



## Field-Programmable Address Decoder (12 × 9)

PLS163

## ORDERING INFORMATION

DESCRIPTION	ORDER CODE
24-pin Plastic DIP 300mil-wide	PLS163N

## ABSOLUTE MAXIMUM RATINGS

SYMBOL	PARAMETER	RATING	UNIT
V <sub>CC</sub>	Supply voltage	+7	V <sub>DC</sub>
V <sub>IN</sub>	Input voltage	+5.5	V <sub>DC</sub>
V <sub>O</sub>	Output voltage Off-state	+5.5	V <sub>DC</sub>
I <sub>IN</sub>	Input current	±30	mA
I <sub>OUT</sub>	Output current	+100	mA
T <sub>A</sub> T <sub>STG</sub>	Temperature range Operating Storage	0 to +75 -65 to +150	°C

DC ELECTRICAL CHARACTERISTICS 0°C ≤ T<sub>A</sub> ≤ +75°C, 4.75 ≤ V<sub>CC</sub> ≤ 5.25V

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ <sup>2</sup>	Max	
<b>Input voltage</b>						
V <sub>IH</sub> V <sub>IL</sub> V <sub>IC</sub>	High <sup>1</sup> Low <sup>1</sup> Clamp <sup>1,3</sup>	V <sub>CC</sub> = Max V <sub>CC</sub> = Min V <sub>CC</sub> = Min, I <sub>IN</sub> = -12mA	2.0		0.8 -1.2	V
<b>Output voltage</b>						
V <sub>OH</sub> V <sub>OL</sub>	High <sup>1,5</sup> Low <sup>1,4</sup>	V <sub>CC</sub> = Min I <sub>OH</sub> = -2mA I <sub>OL</sub> = 15mA	2.4	0.35	0.5	V
<b>Input current</b>						
I <sub>IH</sub> I <sub>IL</sub>	High Low	V <sub>CC</sub> = Max V <sub>IN</sub> = 5.5V V <sub>IN</sub> = 0.45V		< 1 -10	40 -100	μA
<b>Output current</b>						
I <sub>O (OFF)</sub> I <sub>OS</sub>	Hi-Z state <sup>6</sup> Short circuit <sup>3,7</sup>	V <sub>CC</sub> = Max V <sub>OUT</sub> = 5.5V V <sub>OUT</sub> = 0.45V V <sub>OUT</sub> = 0V		1 -1	40 -40 -70	μA mA
I <sub>CC</sub>	V <sub>CC</sub> supply current <sup>8</sup>	V <sub>CC</sub> = Max		120	155	mA
<b>Capacitance</b>						
C <sub>IN</sub> C <sub>OUT</sub>	Input Output <sup>6</sup>	V <sub>CC</sub> = 5.0V V <sub>IN</sub> = 2.0V V <sub>OUT</sub> = 2.0V		8 15		pF

5

## Field-Programmable Address Decoder (12 × 9)

PLS163

**AC ELECTRICAL CHARACTERISTICS**  $0^{\circ}\text{C} \leq T_A \leq +75^{\circ}\text{C}$ ,  $4.75 \leq V_{CC} \leq 5.25\text{V}$ ,  $R_1 = 470\Omega$ ,  $R_2 = 1\text{k}\Omega$ 

SYMBOL	PARAMETER	TO	FROM	TEST CONDITIONS	LIMITS			UNIT
					Min	Typ <sup>2</sup>	Max	
$t_{PD}$	Propagation delay	Output	Input	$C_L = 30\text{pF}$		20	30	ns
$t_{CE}$	Chip enable	Output	Chip enable	$C_L = 30\text{pF}$		20	30	ns
$t_{CD}$	Chip disable	Output	Chip enable	$C_L = 5\text{pF}$		20	30	ns

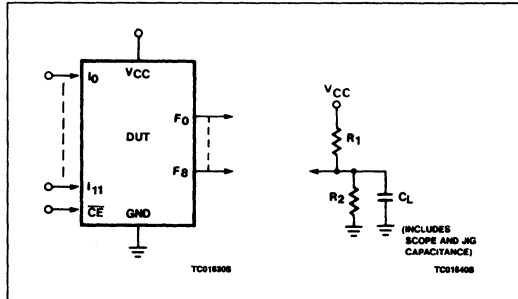
**NOTES:**

1. All voltage values are with respect to network ground terminal.
2. All typical values are at  $V_{CC} = 5\text{V}$ ,  $T_A = +25^{\circ}\text{C}$ .
3. Test each pin one at a time.
4. Measure with a programmed logic condition for which the output under test is at low logic level. Output sink current is supplied through a resistor to  $V_{CC}$ .
5. Measured with  $V_{IL}$  applied to  $\overline{CE}$  and logic high at the output.
6. Measured with  $V_{IH}$  applied to  $\overline{CE}$ .
7. Duration of short circuit should not exceed 1 second.
8.  $I_{CC}$  is measured with the outputs open.

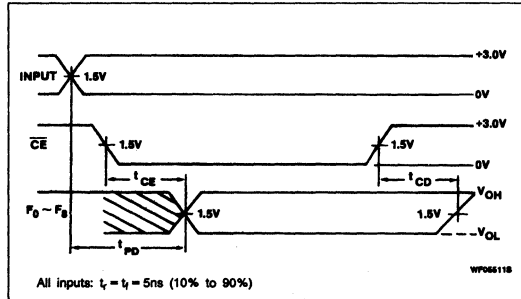
# Field-Programmable Address Decoder (12 × 9)

PLS163

## TEST LOAD CIRCUIT



## VOLTAGE WAVEFORM



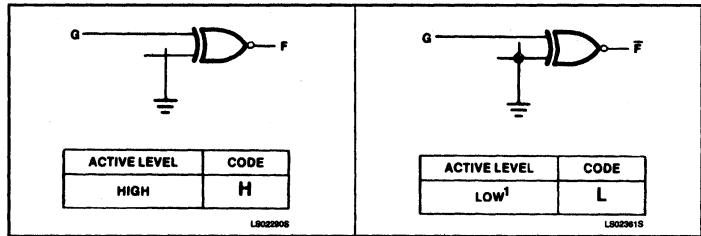
## LOGIC PROGRAMMING

In a virgin device all Ni-Cr links are intact. The FPAD can be programmed by means of Logic Programming equipment.

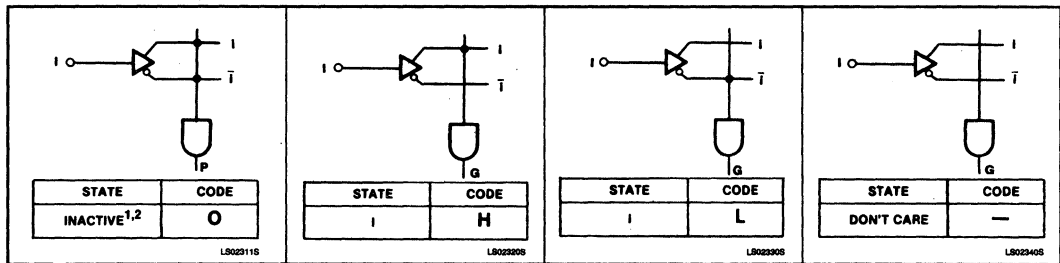
With Logic programming, the AND/EX-OR gate input connections necessary to implement the desired logic function are coded directly from the logic equations using the Program Table on the following page.

In this table, the logic state of variables I and F associated with each gate G<sub>n</sub> is assigned a symbol which results in the proper fusing pattern of corresponding link pairs, defined as follows:

## OUTPUT POLARITY - (F)



## "AND" ARRAY - (I)



### NOTES:

1. This is the initial unprogrammed state of all links.
2. Any gate G<sub>n</sub> will be unconditionally inhibited if both the true and complement fuses of any input (I) are left intact.

## VIRGIN DEVICE

The PLS163 is shipped in an unprogrammed state, characterized by:

1. All P<sub>n</sub> terms are disabled. (Inactive).
2. All P<sub>n</sub> terms are active on all outputs.
3. All outputs are Active-Low.

# Field-Programmable Address Decoder (12 × 9)

PLS163

## FPAD PROGRAM TABLE

CUSTOMER NAME _____ PURCHASE ORDER # _____ SIGNETICS DEVICE # _____ TOTAL NUMBER OF PARTS _____ PROGRAM TABLE # _____	THIS PORTION TO BE COMPLETED BY SIGNETICS CF (XXXX) _____ CUSTOMER SYMBOLIZED PART # _____ DATE RECEIVED _____ COMMENTS _____
---	---

F<sub>0</sub> (16) \_\_\_\_\_ = \_\_\_\_\_  
 F<sub>1</sub> (15) \_\_\_\_\_ = \_\_\_\_\_  
 F<sub>2</sub> (14) \_\_\_\_\_ = \_\_\_\_\_  
 F<sub>3</sub> (13) \_\_\_\_\_ = \_\_\_\_\_  
 F<sub>4</sub> (11) \_\_\_\_\_ = \_\_\_\_\_  
 F<sub>5</sub> (10) \_\_\_\_\_ = \_\_\_\_\_  
 F<sub>6</sub> (9) \_\_\_\_\_ = \_\_\_\_\_  
 F<sub>7</sub> (8) \_\_\_\_\_ = \_\_\_\_\_  
 F<sub>8</sub> (7) \_\_\_\_\_ = \_\_\_\_\_

GATE	INPUT												
	POLARITY	I <sub>11</sub>	I <sub>10</sub>	I <sub>9</sub>	I <sub>8</sub>	I <sub>7</sub>	I <sub>6</sub>	I <sub>5</sub>	I <sub>4</sub>	I <sub>3</sub>	I <sub>2</sub>	I <sub>1</sub>	I <sub>0</sub>
F <sub>0</sub>													
F <sub>1</sub>													
F <sub>2</sub>													
F <sub>3</sub>													
F <sub>4</sub>													
F <sub>5</sub>													
F <sub>6</sub>													
F <sub>7</sub>													
F <sub>8</sub>													
PIN NO.	18	19	20	21	22	23	1	2	3	4	5	6	
VARIABLE NAME													

**NOTES**

- The FPAD is shipped with all links intact. Thus a background of entries corresponding to states of virgin links exists in the table, shown BLANK for clarity.
- Unused I bits are normally programmed Don't Care (—).
- Unused Gates can be left blank.

AND		CONTROL	
INACTIVE : 0	I : H	HIGH : H	
T : L	Don't Care : —	LOW : L	
(I)		(POL.)	

## PLS167 Field-Programmable Logic Sequencer (14 × 48 × 6)

Signetics Programmable Logic  
Product Specification

### Application Specific Products • Series 24

#### DESCRIPTION

The PLS167 is a bipolar, programmable state machine of the Mealy type. The Field Programmable Logic Sequencer (FPLS) contains logic AND-OR gate arrays with user programmable connections which control the inputs of on-chip State and Output Registers. These consist respectively of 8  $Q_P$ , and 4  $Q_I$  edge-triggered, clocked S/R flip-flops, with an asynchronous preset option.

All flip-flops are unconditionally preset to "1" during power turn-on.

The AND Array combines 14 external inputs,  $I_0-13$ , with 8 internal inputs,  $P_0-7$ , fed back from the State Register to form up to 48 transition terms (AND terms). In addition,  $P_0$  and  $P_1$  of the internal State Register are brought off-chip to allow extending the Output Register to 6 bits, if so desired.

All transition terms can include True, False, or Don't Care states of the controlling variables, and are merged in the OR Array to issue next-state and next-output commands to their respective registers to the Low-to-High transition of the Clock pulse.

Both True and Complement transition terms can be generated by optional use of the internal variable (C) from the Complement Array. Also, if desired, the Preset input can be converted to Output-

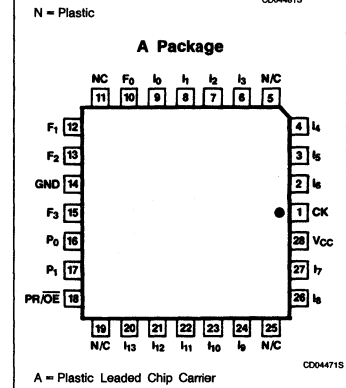
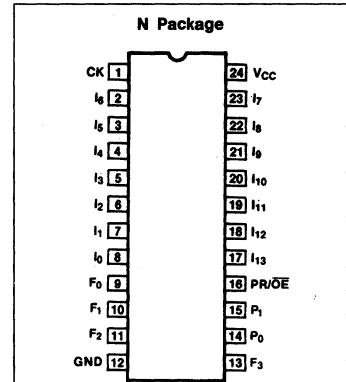
Enable function, as an additional user programmable option.

Order codes are contained on the pages following.

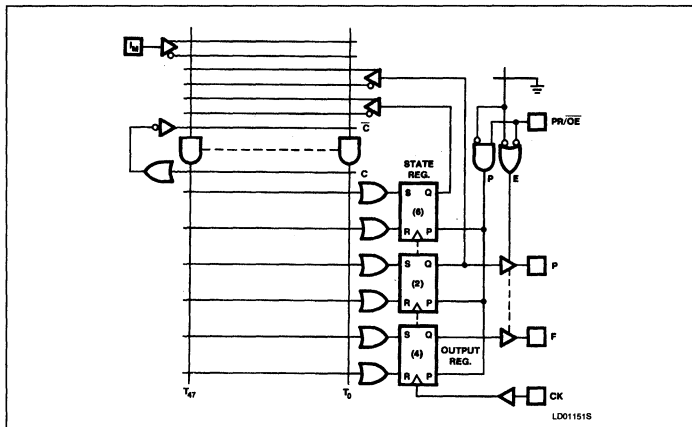
#### FEATURES

- Field-Programmable (Ni-Cr link)
- 14 True/Complement buffered inputs
- 48 programmable AND gates
- 25 programmable OR gates
- 8-bit State Register
- 2-bit shared State/Output Register
- 4-bit Output Register
- Transition Complement Array
- Programmable Asynchronous Preset/Output Enable
- Positive edge-triggered clock
- Power-on preset to logic "1" of all registers
- Automatic logic "HOLD" state via S/R flip-flops
- On-chip Test Array
- $f_{MAX}$ : 13.9MHz
- Power: 600mW (typ.)
- TTL compatible
- Tri-state outputs
- Single +5V supply
- 300mil-wide 24-pin DIP

#### PIN CONFIGURATIONS



#### FUNCTIONAL DIAGRAM



#### APPLICATIONS

- Interface protocols
- Sequence detectors
- Peripheral controllers
- Timing generators
- Sequential circuits
- Security locking systems



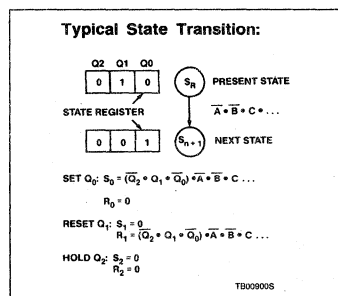
# Field-Programmable Logic Sequencer (14 × 48 × 6)

PLS167

## PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION	POLARITY
1	CK	<b>Clock:</b> The Clock input to the State and Output Registers. A Low-to-High transition on this line is necessary to update the contents of both registers.	Active-High
2-7 17-23	I <sub>1-13</sub>	<b>Logic Inputs:</b> The 13 external inputs to the AND array used to program jump conditions between machine states, as determined by a given logic sequence.	Active-High/Low
8	I <sub>0</sub>	<b>Logic/Diagnostic Input:</b> A 14th external logic input to the AND array, as above, when exercised with standard TTL levels. When I <sub>0</sub> is held at +10V, device outputs F <sub>0-3</sub> and P <sub>0-1</sub> reflect the contents of State Register bits P <sub>2-7</sub> (see Diagnostic Output Mode diagram). The contents of flip-flops P <sub>0-1</sub> and F <sub>0-3</sub> remain unaltered.	Active-High/Low
9-11 13	F <sub>0-3</sub>	<b>Logic/Diagnostic Outputs:</b> Four device outputs which normally reflect the contents of Output Register bits Q <sub>0-3</sub> , when enabled. When I <sub>0</sub> is held at +10V, F <sub>0-3</sub> = (P <sub>2-5</sub> ).	Active-High
14-15	P <sub>0-1</sub>	<b>Logic/Diagnostic Outputs:</b> Two register bits with shared function as least significant State Register bits, or most significant Output Register bits. When I <sub>0</sub> is held at +10V, P <sub>0-1</sub> = (P <sub>6-7</sub> ).	Active-High
16	PR/ $\overline{OE}$	<b>Preset or Output Enable Input:</b> A user programmable function: • <b>Preset:</b> Provides an asynchronous preset to logic "1" of all State and Output Register bits. Preset overrides Clock, and when held High, clocking is inhibited and P <sub>0-1</sub> and F <sub>0-3</sub> are High. Normal clocking resumes with the first full clock pulse following a High-to-Low clock transition, after Preset goes Low. • <b>Output Enable:</b> Provides an Output Enable function to all output buffers.	Active-High (H)  Active-Low (L)

## LOGIC FUNCTION



## TRUTH TABLE 1, 2, 3, 4, 5, 6

V <sub>CC</sub>	OPTION		I <sub>0</sub>	CK	S	R	Q <sub>P/F</sub>	F	
	PR	$\overline{OE}$							
+5V	H		*	X	X	X	H	H	
	L		+10V	X	X	X	Q <sub>n</sub>	(Q <sub>P</sub> ) <sub>n</sub>	
	L		X	X	X	X	Q <sub>n</sub>	(Q <sub>F</sub> ) <sub>n</sub>	
		H	*	X	X	X	Q <sub>n</sub>	Hi-Z	
		L	+10V	X	X	X	Q <sub>n</sub>	(Q <sub>P</sub> ) <sub>n</sub>	
		L	X	X	X	X	Q <sub>n</sub>	(Q <sub>F</sub> ) <sub>n</sub>	
			L	X	↑	L	L	Q <sub>n</sub>	(Q <sub>F</sub> ) <sub>n</sub>
			L	X	↑	L	H	L	L
		L	X	↑	H	L	L	H	
		L	X	↑	H	H	IND.	IND.	
↑	X	X	X	X	X	X	H		

### NOTES:

- Positive Logic:  
S/R = T<sub>0</sub> + T<sub>1</sub> + T<sub>2</sub> + ... + T<sub>47</sub>  
T<sub>n</sub> = C(I<sub>0</sub> I<sub>1</sub> I<sub>2</sub>...) (P<sub>0</sub> P<sub>1</sub>...P<sub>7</sub>)
- Either Preset (Active-High) or Output Enable (Active-Low) are available, but not both. The desired function is a user programmable option.
- ↑ denotes transition from Low to High level.
- R = S = High is an illegal input condition.
- \* = H/L/+10V
- X = Don't Care (≤ 5.5V)

## VIRGIN STATE

A factory shipped virgin device contains all fusible links intact, such that:

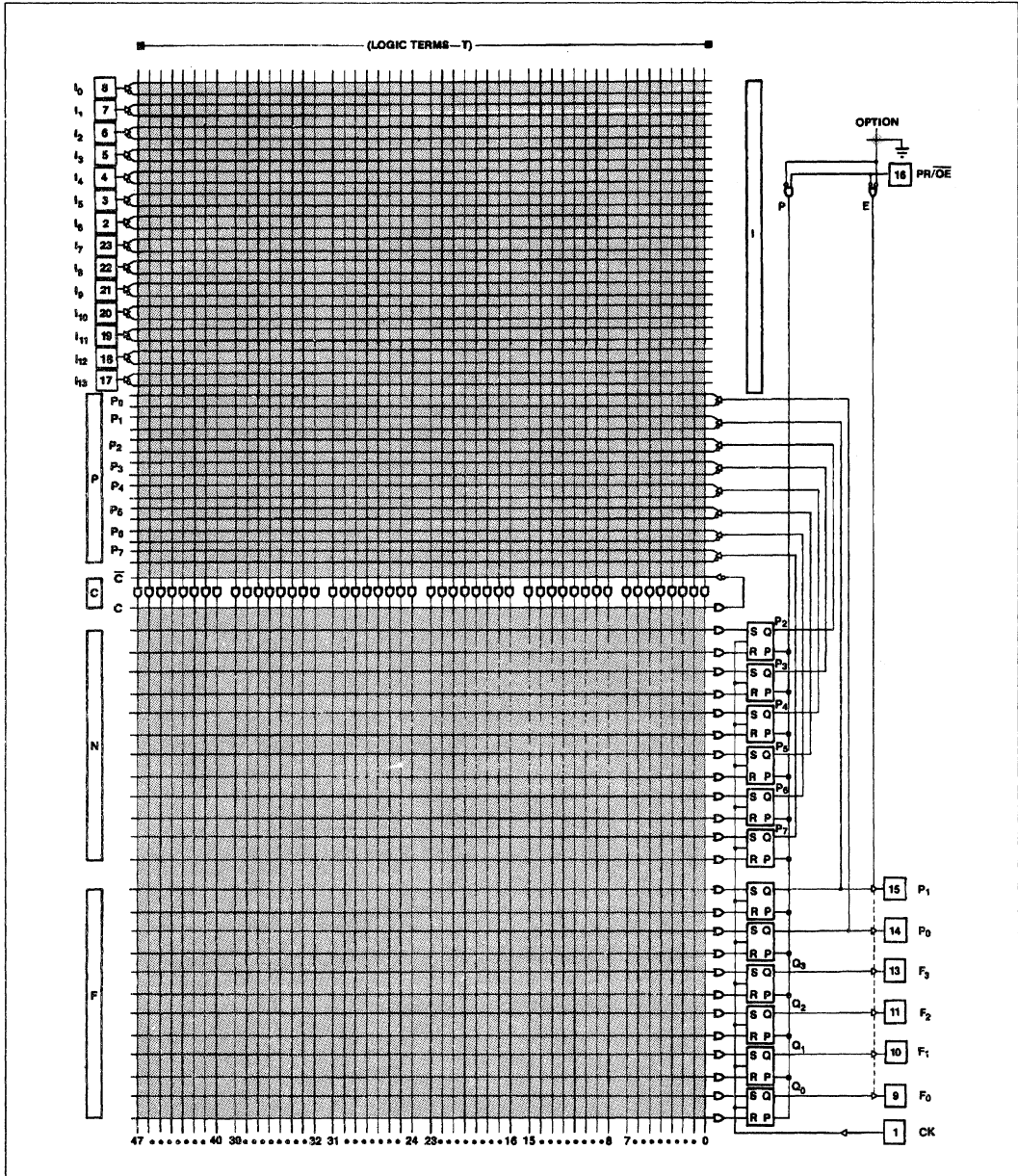
- PR/ $\overline{OE}$  option is set to PR. Thus, all outputs will be at "1", as preset by initial power-up procedure.
- All transition terms are disabled (0).
- All S/R flip-flop inputs are disabled (0).
- The device can be clocked via a Test Array pre-programmed with a standard test pattern.

NOTE: The Test Array pattern MUST be deleted before incorporating a user program. This is accomplished automatically by any Signetics qualified programming equipment.

# Field-Programmable Logic Sequencer (14 × 48 × 6)

PLS167

## FPLS LOGIC DIAGRAM



- NOTES:**
1. All AND gate inputs with a blown link float to a logic "1".
  2. All OR gate inputs with a blown fuse float to logic "0".
  3. Programmable connection.

## Field-Programmable Logic Sequencer (14 × 48 × 6)

PLS167

## ORDERING INFORMATION

DESCRIPTION	ORDER CODE
24-pin Plastic DIP 300mil-wide	PLS167N
28-pin Plastic Leaded Chip Carrier	PLS167A

## THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS<sup>1</sup>

SYMBOL	PARAMETER	RATING		UNIT
		Min	Max	
V <sub>CC</sub>	Supply voltage		+7	V <sub>DC</sub>
V <sub>IN</sub>	Input voltage		+5.5	V <sub>DC</sub>
V <sub>OUT</sub>	Output voltage		+5.5	V <sub>DC</sub>
I <sub>IN</sub>	Input currents	-30	+30	mA
I <sub>OUT</sub>	Output currents		+100	mA
T <sub>A</sub> T <sub>STG</sub>	Temperature range Operating Storage	0 -65	+75 +150	°C

## Field-Programmable Logic Sequencer (14 × 48 × 6)

PLS167

DC ELECTRICAL CHARACTERISTICS  $0^{\circ}\text{C} \leq T_A \leq 75^{\circ}\text{C}$ ,  $4.75\text{V} \leq V_{CC} \leq 5.25\text{V}$ 

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ <sup>2</sup>	Max	
<b>Input voltage<sup>3</sup></b>						
V <sub>IH</sub> V <sub>IL</sub> V <sub>IC</sub>	High Low Clamp <sup>4</sup>	V <sub>CC</sub> = Max V <sub>CC</sub> = Min V <sub>CC</sub> = Min, I <sub>IN</sub> = -12mA	2		0.8 -1.2	V
<b>Output voltage<sup>3</sup></b>						
V <sub>OH</sub> V <sub>OL</sub>	High <sup>5</sup> Low <sup>6</sup>	V <sub>CC</sub> = Min I <sub>OH</sub> = -2mA I <sub>OL</sub> = 9.6mA	2.4	0.35	0.45	V
<b>Input current</b>						
I <sub>IH</sub> I <sub>IL</sub> I <sub>IL</sub>	High Low Low (CK input)	V <sub>IN</sub> = 5.5V V <sub>IN</sub> = 0.45V V <sub>IN</sub> = 0.45V		< 1 -10 -50	25 -100 -250	μA
<b>Output current</b>						
I <sub>O(OFF)</sub> I <sub>OS</sub>	Hi-Z state <sup>7</sup> Short circuit <sup>4,8</sup>	V <sub>CC</sub> = Max V <sub>OUT</sub> = 5.5V V <sub>OUT</sub> = 0.45V V <sub>OUT</sub> = 0V		1 -1	40 -40 -70	μA mA
I <sub>CC</sub>	V <sub>CC</sub> supply current <sup>9</sup>	V <sub>CC</sub> = Max		120	180	mA
<b>Capacitance<sup>7</sup></b>						
C <sub>IN</sub> C <sub>OUT</sub>	Input Output	V <sub>CC</sub> = 5.0V V <sub>IN</sub> = 2.0V V <sub>OUT</sub> = 2.0V		8 10		pF

## NOTES:

- Stresses above those listed under "Absolute Maximum Ratings" may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.
- All typical values are at V<sub>CC</sub> = 5V, T<sub>A</sub> = +25°C.
- All voltage values are with respect to network ground terminal.
- Test one at a time.
- Measured with V<sub>IL</sub> applied to  $\overline{OE}$  and a logic high stored, or with V<sub>IH</sub> applied to PR.
- Measured with a programmed logic condition for which the output is at a low logic level, and V<sub>IL</sub> applied to PR/ $\overline{OE}$ . Output sink current is supplied through a resistor to V<sub>CC</sub>.
- Measured with V<sub>IH</sub> applied to PR/ $\overline{OE}$ .
- Duration of short circuit should not exceed 1 second.
- I<sub>CC</sub> is measured with the PR/ $\overline{OE}$  input grounded, all other inputs at 4.5V and the outputs open.

## Field-Programmable Logic Sequencer (14 × 48 × 6)

PLS167

**AC ELECTRICAL CHARACTERISTICS**  $R_1 = 470\Omega$ ,  $R_2 = 1k\Omega$ ,  $C_L = 30pF$ ,  $0^\circ C \leq T_A \leq +75^\circ C$ ,  $4.75V \leq V_{CC} \leq 5.25V$ 

SYMBOL	PARAMETER	TO	FROM	LIMITS			UNIT
				Min	Typ <sup>1</sup>	Max	
<b>Pulse width<sup>3</sup></b>							
$t_{CKH}$	Clock <sup>2</sup> high	CK -	CK +	25	15		ns
$t_{CKL}$	Clock low	CK +	CK -	25	15		
$t_{CKP1B}$	Period (without Complement Array)	CK +	CK +	80	40		
$t_{CKP2B}$	Period (with Complement Array)	CK +	CK +	120	60		
$t_{PRH}$	Preset pulse	PR -	PR +	25	15		
<b>Setup time<sup>3</sup></b>							
$t_{S1A}$	Input	CK +	Input $\pm$	60			ns
$t_{S1B}$	Input	CK +	Input $\pm$	50			
$t_{S1C}$	Input	CK +	Input $\pm$	42			
$t_{S2A}$	Input (through Complement Array)	CK +	Input $\pm$	90			
$t_{S2B}$	Input (through Complement Array)	CK +	Input	80			
$t_{S2C}$	Input (through Complement Array)	CK +	Input	72			
$t_{VS}$	Power-on preset	CK -	$V_{CC} +$	0	-10		
$t_{PRS}$	Preset	CK -	PR -	0	-10		
<b>Hold time</b>							
$t_H$	Input	Input $\pm$	CK +	5	-10		ns
<b>Propagation delay</b>							
$t_{CKO}$	Clock	Output $\pm$	CK +		15	30	ns
$t_{OE}$	Output enable	Output -	OE -		20	30	
$t_{OD}$	Output disable	Output +	OE +		20	30	
$t_{PR}$	Preset	Output +	PR +		18	30	
$t_{PPR}$	Power-on preset	Output +	$V_{CC} +$		0	10	
<b>Frequency of operation<sup>3</sup></b>							
$f_{MAXC}$	Without Complement Array					13.9	MHz
$f_{MAXC}$	With Complement Array					9.8	

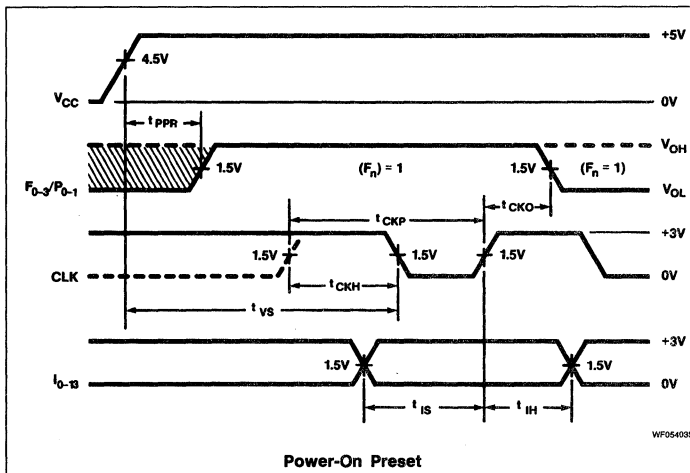
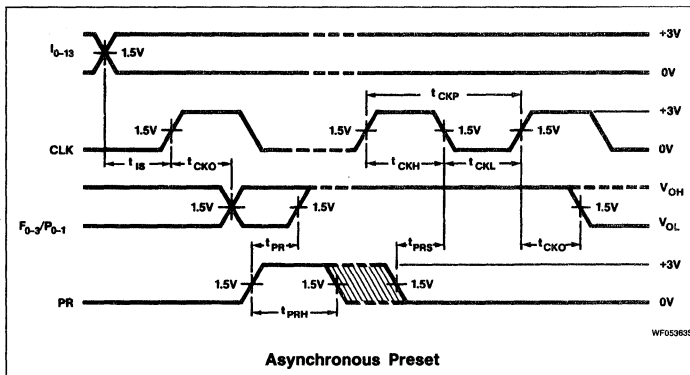
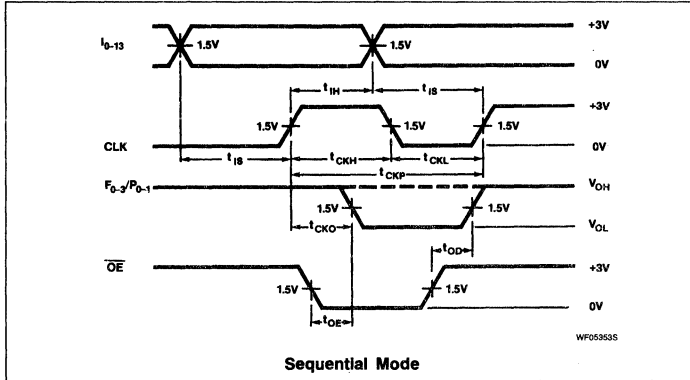
**NOTES:**

- All typical values are at  $V_{CC} = 5V$ ,  $T_A = +25^\circ C$ .
- To prevent spurious clocking, clock rise time (10% - 90%)  $\leq 30ns$ .
- See "Speed vs. OR Loading" diagrams.

# Field-Programmable Logic Sequencer (14 × 48 × 6)

PLS167

## TIMING DIAGRAMS



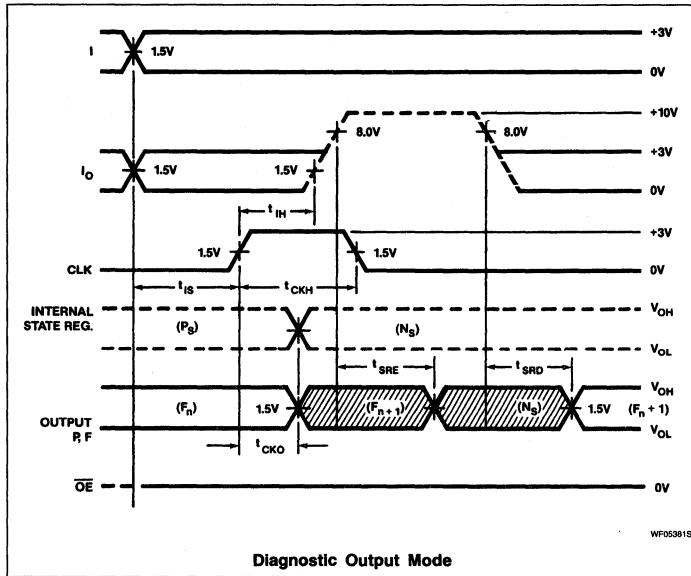
## TIMING DEFINITIONS

- $t_{CKH}$  Width of input clock pulse.
- $t_{CKL}$  Interval between clock pulses.
- $t_{CKP1}$  Clock period – when not using Complement array.
- $t_{IS1}$  Required delay between beginning of valid input and positive transition of clock.
- $t_{CKP2}$  Clock period – when using complement array.
- $t_{IS2}$  Required delay between beginning of valid input and positive transition of clock, when using optional Complement Array (two passes necessary through the AND array).
- $t_{VS}$  Required delay between  $V_{CC}$  (after power-on) and negative transition of clock preceding first reliable clock pulse.
- $t_{PRS}$  Required delay between negative transition of asynchronous Preset and negative transition of clock preceding first reliable clock pulse.
- $t_{IH}$  Required delay between positive transition of clock and end of valid input data.
- $t_{CKO}$  Delay between positive transition of clock and when Outputs become valid (with PR/ $\overline{OE}$  low).
- $t_{OE}$  Delay between beginning of Output Enable Low and when Outputs become valid.
- $t_{OD}$  Delay between beginning of Output Enable High and when Outputs are in the off state.
- $t_{SRE}$  Delay between input  $I_0$  transition to Diagnostic mode and when the Outputs reflect the contents of the State Register.
- $t_{SRD}$  Delay between input  $I_0$  transition to Logic mode and when the Outputs reflect the contents of the Output Register.
- $t_{PR}$  Delay between positive transition of Preset and when Outputs become valid at "1".
- $t_{PPR}$  Delay between  $V_{CC}$  (after power-on) and when Outputs become preset at "1".
- $t_{PRH}$  Width of preset input pulse.
- $f_{MAX}$  Maximum clock frequency.

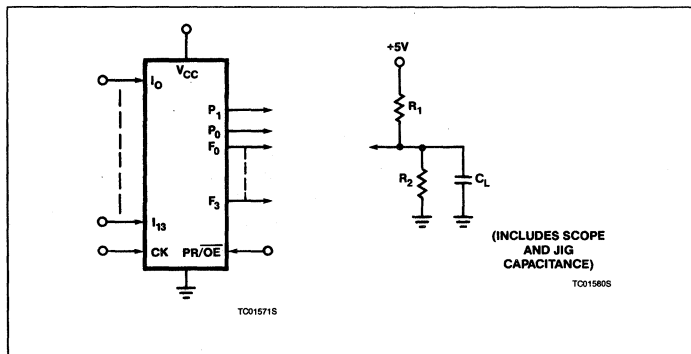
# Field-Programmable Logic Sequencer (14 × 48 × 6)

PLS167

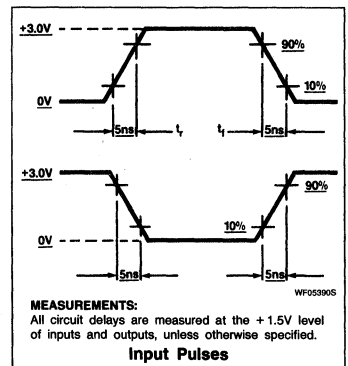
## TIMING DIAGRAMS (Continued)



## TEST LOAD CIRCUIT



## VOLTAGE WAVEFORM



# Field-Programmable Logic Sequencer (14 × 48 × 6)

PLS167

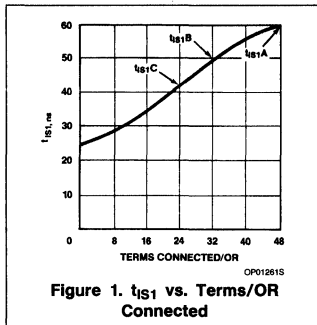
## SPEED VS. "OR" LOADING

The maximum frequency at which the FPLS can be clocked while operating in *sequential mode* is given by:

$$(1/f_{MAX}) = t_{CY} = t_{IS} + t_{CKO}$$

This frequency depends on the number of transition terms  $T_n$  used. Having all 48 terms connected in the AND array does not appreciably impact performance; but the number of terms connected to each OR line affects  $t_{IS}$ , due to capacitive loading. The effect of this loading can be seen in Figure 1, showing the variation of  $t_{IS1}$  with the number of terms connected per OR.

The AC electrical characteristics contain three limits for the parameters  $t_{IS1}$  and  $t_{IS2}$ . The first,  $t_{IS1A}$  is guaranteed for a device with 48 terms connected to any OR line.  $t_{IS1B}$  is guaranteed for a device with 32 terms connected to any OR line. And  $t_{IS1C}$  is guaranteed for a device with 24 terms connected to any OR line.



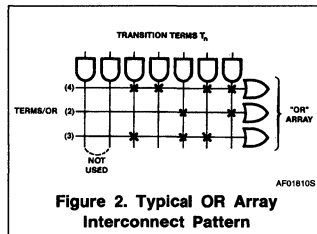
The Three other entries in the AC table,  $t_{IS2}$  A, B, and C are corresponding 48, 32, and 24 term limits when using the on-chip Complement Array.

The worst case  $t_{IS}$  for a given application can be determined by identifying the OR line with the maximum number of  $T_n$  connections. This can be done by referring to

the interconnect pattern in the FPLS logic diagram, typically illustrated in Figure 2, or by counting the maximum number of "H" or "L" entries in one of the columns of the device Program Table.

This number plotted on the curve in Figure 1 will yield the worst case  $t_{IS}$  and, by implication, the maximum clocking frequency for reliable operation.

*Note that for maximum speed all UNUSED transition terms should be disconnected from the OR array.*



**Figure 2. Typical OR Array Interconnect Pattern**



# Field-Programmable Logic Sequencer (14 × 48 × 6)

PLS167

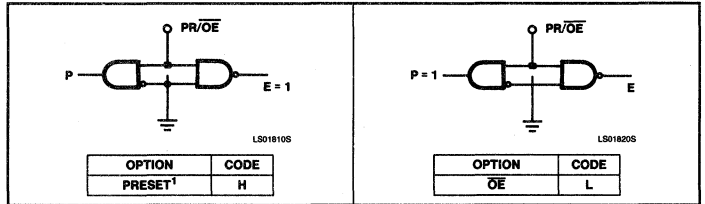
## LOGIC PROGRAMMING

The FPLS can be programmed by means of Logic programming equipment.

With Logic programming, the AND/OR gate input connections necessary to implement the desired logic function are coded directly from the State Diagram using the Program Table on the following page.

In this table, the logic state or action of control variables C, I, P, N, and F, associated with each Transition Term  $T_n$ , is assigned a symbol which results in the proper fusing pattern of corresponding links, defined as follows:

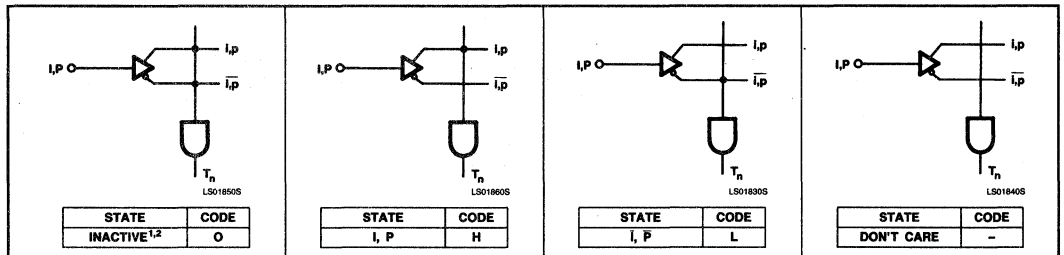
## PRESET/ÖE OPTION - (P/E)



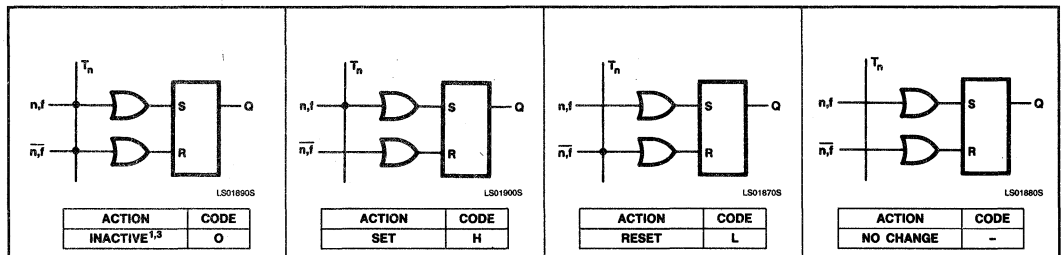
### PROGRAMMING:

The PLS167 has a power-up preset feature. This feature insures that the device will power-up in a known state with all register elements (state and output register) at a logic High (H). When programming the device it is important to realize this is the initial state of the device. You *must* provide a next state jump if you do not wish to use all Highs (H) as the present state.

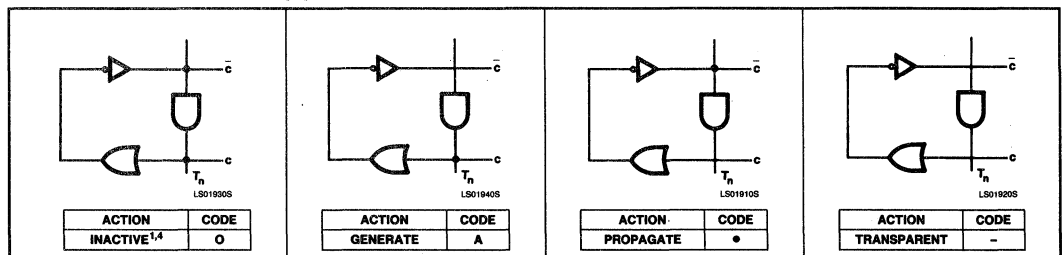
## "AND" ARRAY - (I), (P)



## "OR" ARRAY - (N), (F)



## "COMPLEMENT" ARRAY - (C)



### NOTES:

1. This is the initial unprogrammed state of all links.
2. Any gate  $T_n$  will be unconditionally inhibited if both the true and complement of any input (I, P) are left intact.
3. To prevent simultaneous Set and Reset flip-flop commands, this state is not allowed for N and F link pairs coupled to active gates  $T_n$  (see flip-flop truth tables).
4. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates  $T_n$ .



# Field-Programmable Logic Sequencer (14 × 48 × 6)

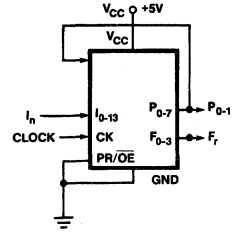
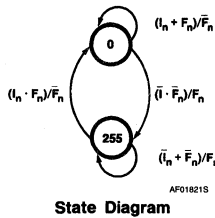
PLS167

## TEST ARRAY

The FPLS may be subjected to AC and DC parametric tests prior to programming via an on-chip test array.

The array consists of test transition terms 48 and 49, factory programmed as shown below.

Testing is accomplished by clocking the FPLS and applying the proper input sequence to  $I_{0-13}$  as shown in the test circuit timing diagram.



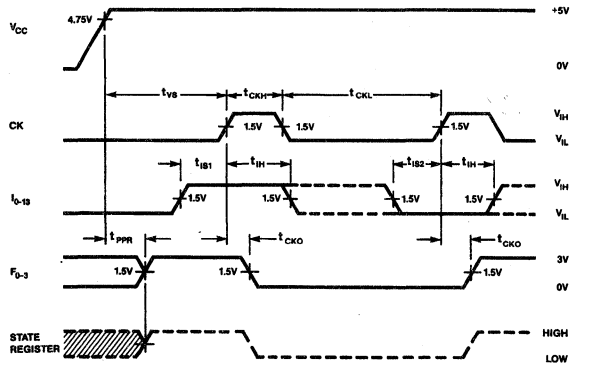
FPLS Under Test

TERM	AND																							
	C	INPUT (Im)												PRESENT STATE (Ps)										
		1	1	1	1	9	8	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	
48	A	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H		
49	●	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L		

OPTION (P/E)		H											
OR													
NEXT STATE (Ns)										OUTPUT (Fr)			
7	6	5	4	3	2	1	0	3	2	1	0		
L	L	L	L	L	L	L	L	L	L	L	L		
H	H	H	H	H	H	H	H	H	H	H	H		

Test Array Program

Both terms 48 and 49 must be deleted during user programming to avoid interfering with the desired logic function. This is accomplished automatically by any of Signetics' qualified programming equipment.



Test Circuit Timing Diagram

TERM	AND																							
	C	INPUT (Im)												PRESENT STATE (Ps)										
		1	1	1	1	9	8	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	
48	-	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H		
49	●	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L		

OPTION (P/E)		H											
OR													
NEXT STATE (Ns)										OUTPUT (Fr)			
7	6	5	4	3	2	1	0	3	2	1	0		
-	-	-	-	-	-	-	-	-	-	-	-		
-	-	-	-	-	-	-	-	-	-	-	-		

Test Array Deleted

## PLS167A Field-Programmable Logic Sequencer (14 × 48 × 6)

Signetics Programmable Logic  
Product Specification

### Application Specific Products • Series 24

#### DESCRIPTION

The PLS167A is a bipolar, programmable state machine of the Mealy type. The Field-Programmable Logic Sequencer (FPLS) contains logic AND-OR gate arrays with user programmable connections which control the inputs of on-chip State and Output Registers. These consist respectively of 8  $Q_D$ , and 4  $Q_I$  edge-triggered, clocked S/R flip-flops, with an Asynchronous Preset Option.

All flip-flops are unconditionally preset to "1" during power turn-on.

The AND array combines 14 external inputs,  $I_0$ - $I_3$ , with 8 internal inputs,  $P_0$ - $P_7$ , fed back from the State Register to form up to 48 transition terms (AND terms). In addition,  $P_0$  and  $P_1$  of the internal State Register are brought off-chip to allow extending the Output Register to 6 bits, if so desired.

All transition terms can include True, False, or Don't Care states of the controlling variables, and are merged in the OR array to issue next-state and next-output commands to their respective registers on the Low-to-High transition of the Clock pulse.

Both True and Complement transition terms can be generated by optional use of the internal variable (C) from the Complement Array. Also, if desired, the Preset input can be converted to output-

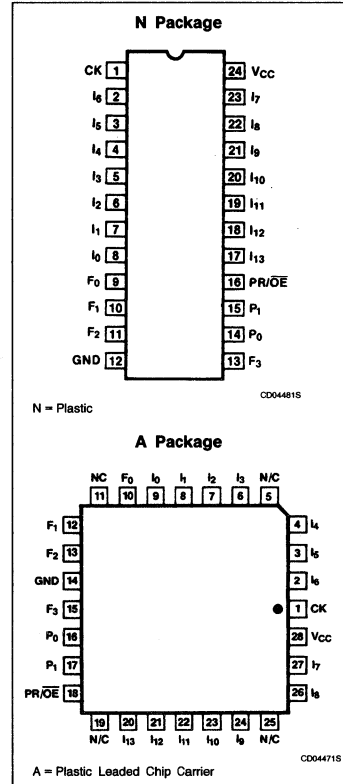
enable function, as an additional user programmable option.

Order codes are contained on the pages following.

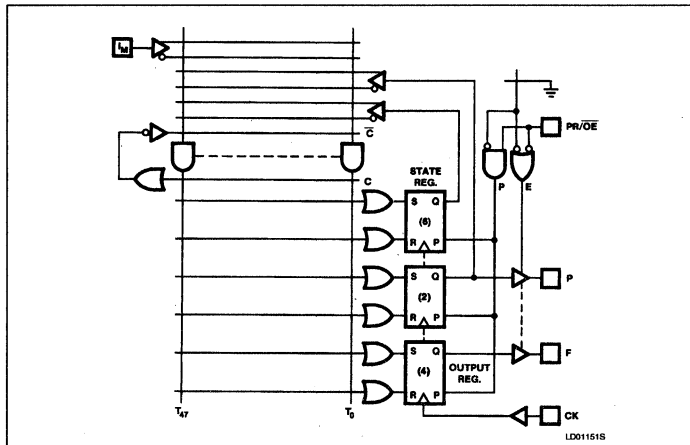
#### FEATURES

- Field-Programmable (Ni-Cr link)
- 14 True/Complement buffered inputs
- 48 programmable AND gates
- 25 programmable OR gates
- 8-bit State Register
- 2-bit shared State/Output Register
- 4-bit Output Register
- Transition Complement Array
- Programmable Asynchronous Preset/Output Enable
- Positive edge-triggered clock
- Power-on preset to logic "1" of all registers
- Automatic logic "HOLD" state via S/R flip-flops
- On-chip Test Array
- $f_{MAX}$ : 20MHz
- Power: 600mW (typ.)
- TTL compatible
- Tri-state outputs
- Single +5V supply
- 300mil-wide 24-pin DIP

#### PIN CONFIGURATIONS



#### FUNCTIONAL DIAGRAM



#### APPLICATIONS

- Interface protocols
- Sequence detectors
- Peripheral controllers
- Timing generators
- Sequential circuits
- Security locking systems

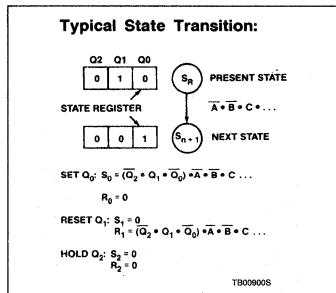
# Field-Programmable Logic Sequencer (14 × 48 × 6)

PLS167A

## PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION	POLARITY
1	CK	<b>Clock:</b> The Clock input to the State and Output Registers. A low-to-high transition on this line is necessary to update the contents of both registers.	Active-High
2 - 7 17 - 23	I <sub>1-13</sub>	<b>Logic Inputs:</b> The 13 external inputs to the AND array used to program jump conditions between machine states, as determined by a given logic sequence.	Active-High/Low
8	I <sub>0</sub>	<b>Logic/Diagnostic Input:</b> A 14th external logic input to the AND array, as above, when exercised with standard TTL levels. When I <sub>0</sub> is held at +10V, device outputs F <sub>0-3</sub> and P <sub>0-1</sub> reflect the contents of state register bits P <sub>2-7</sub> (see Diagnostic Output Mode diagram). The contents of flip-flops P <sub>0-1</sub> and F <sub>0-3</sub> remain unaltered.	Active-High/Low
9 - 11 13	F <sub>0-3</sub>	<b>Logic/Diagnostic Outputs:</b> Four device outputs which normally reflect the contents of Output Register bits Q <sub>0-3</sub> , when enabled. When I <sub>0</sub> is held at +10V, F <sub>0-3</sub> = (P <sub>2-5</sub> ).	Active-High
14 - 15	P <sub>0-1</sub>	<b>Logic/Diagnostic Outputs:</b> Two register bits with shared function as least significant State Register bits, or most significant Output Register bits. When I <sub>0</sub> is held at +10V, P <sub>0-1</sub> = (P <sub>6-7</sub> ).	Active-High
16	PR/ $\overline{OE}$	<b>Preset or Output Enable Input:</b> A user programmable function: • <b>Preset:</b> Provides an Asynchronous Preset to logic "1" of all State and Output Register bits. Preset overrides Clock, and when held High, clocking is inhibited and P <sub>0-1</sub> and F <sub>0-3</sub> are High. Normal clocking resumes with the first full clock pulse following a High-to-Low clock transition, after Preset goes Low. • <b>Output Enable:</b> Provides an Output Enable function to all output buffers.	Active High (H)  Active-Low (L)

## LOGIC FUNCTION



## TRUTH TABLE<sup>1, 2, 3, 4, 5, 6</sup>

V <sub>CC</sub>	OPTION		I <sub>0</sub>	CK	S	R	Q <sub>P/F</sub>	F
	PR	$\overline{OE}$						
+5V	H	X	*	X	X	X	H	H
	L	X	+10V	X	X	X	Q <sub>n</sub>	(Q <sub>P</sub> ) <sub>n</sub>
	L	X	X	X	X	X	Q <sub>n</sub>	(Q <sub>F</sub> ) <sub>n</sub>
	X	H	*	X	X	X	Q <sub>n</sub>	Hi-Z
	X	L	+10V	X	X	X	Q <sub>n</sub>	(Q <sub>P</sub> ) <sub>n</sub>
	X	L	X	X	X	X	Q <sub>n</sub>	(Q <sub>F</sub> ) <sub>n</sub>
↑	X	X	L	↑	L	L	Q <sub>n</sub>	(Q <sub>F</sub> ) <sub>n</sub>
			L	↑	L	H	L	L
			L	↑	H	L	L	H
			L	↑	H	H	IND.	IND.
↑	X	X	X	X	X	H	X	

### NOTES:

- Positive Logic:  
 $S/R = T_0 + T_1 + T_2 + \dots + T_{47}$   
 $T_n = C(I_0 \ I_1 \ I_2 \dots) (P_0 \ P_1 \dots P_7)$
- Either Preset (Active-High) or Output Enable (Active-Low) are available, but not both. The desired function is a user programmable option.
- ↑ denotes transition from Low-to-High level.
- R = S = High is an illegal input condition.
- \* = H/L/+10V
- X = Don't care ( $\leq 5.5V$ )

## VIRGIN STATE

A factory shipped virgin device contains all fusible links intact, such that:

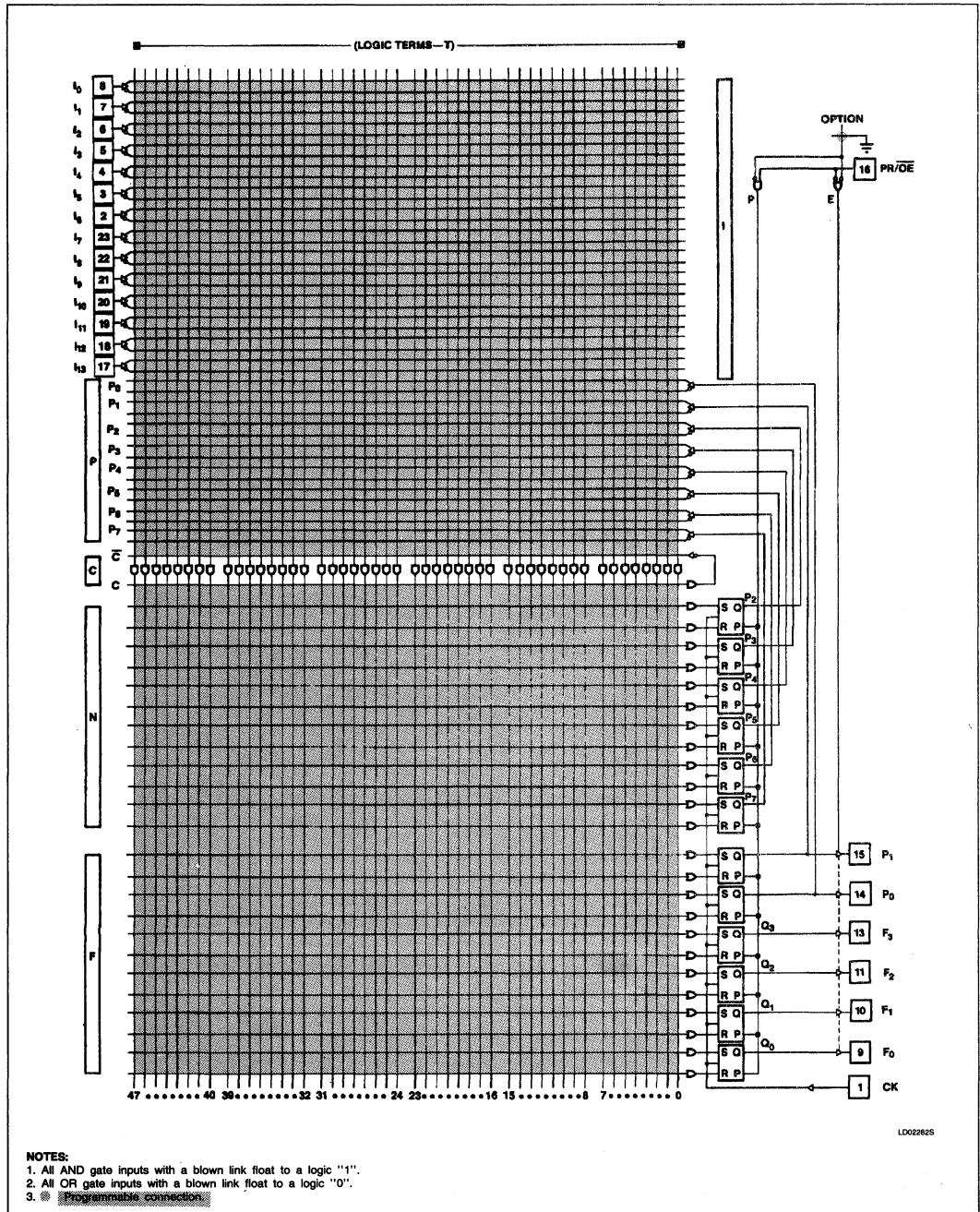
- PR/ $\overline{OE}$  option is set to PR. Thus, all outputs will be at "1", as preset by initial power-up procedure.
- All transition terms are disabled (0).
- All S/R flip-flop inputs are disabled (0).
- The device can be clocked via a Test Array pre-programmed with a standard test pattern.

NOTE: The Test Array pattern MUST be deleted before incorporating a user program. This is accomplished automatically by any Signetics qualified programming equipment.

# Field-Programmable Logic Sequencer (14 × 48 × 6)

## PLS167A

### FPLS LOGIC DIAGRAM



- NOTES:**
- 1. All AND gate inputs with a blown link float to a logic "1".
  - 2. All OR gate inputs with a blown link float to a logic "0".
  - 3. \* Programmable connection.

LD022825

## Field-Programmable Logic Sequencer (14 × 48 × 6)

PLS167A

## ORDERING INFORMATION

DESCRIPTION	ORDER CODE
24-pin Plastic DIP 300mil-wide	PLS167AN
28-pin Plastic Leaded Chip Carrier	PLS167AA

## THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS<sup>1</sup>

SYMBOL	PARAMETER	RATING		UNIT
		Min	Max	
V <sub>CC</sub>	Supply voltage		+7	V <sub>DC</sub>
V <sub>IN</sub>	Input voltage		+5.5	V <sub>DC</sub>
V <sub>OUT</sub>	Output voltage		+5.5	V <sub>DC</sub>
I <sub>IN</sub>	Input currents	-30	+30	mA
I <sub>OUT</sub>	Output currents		+100	mA
T <sub>A</sub> T <sub>STG</sub>	Temperature range Operating Storage	0 -65	+75 +150	°C

# Field-Programmable Logic Sequencer (14 × 48 × 6)

PLS167A

## DC ELECTRICAL CHARACTERISTICS $0^{\circ}\text{C} \leq T_A \leq 75^{\circ}\text{C}$ , $4.75\text{V} \leq V_{CC} \leq 5.25\text{V}$

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ <sup>2</sup>	Max	
<b>Input voltage<sup>3</sup></b>						
V <sub>IH</sub> V <sub>IL</sub> V <sub>IC</sub>	High Low Clamp <sup>4</sup>	V <sub>CC</sub> = Max V <sub>CC</sub> = Min V <sub>CC</sub> = Min, I <sub>IN</sub> = -12mA	2		0.8 -1.2	V
<b>Output voltage<sup>3</sup></b>						
V <sub>OH</sub> V <sub>OL</sub>	High <sup>5</sup> Low <sup>6</sup>	V <sub>CC</sub> = Min I <sub>OH</sub> = -2mA I <sub>OL</sub> = 9.6mA	2.4	0.35	0.45	V
<b>Input current</b>						
I <sub>IH</sub> I <sub>IL</sub> I <sub>IL</sub>	High Low Low (CK input)	V <sub>IN</sub> = 5.5V V <sub>IN</sub> = 0.45V V <sub>IN</sub> = 0.45V		< 1 -10 -50	25 -100 -250	μA
<b>Output current</b>						
I <sub>O(OFF)</sub> I <sub>OS</sub>	Hi-Z state <sup>7</sup> Short circuit <sup>4,8</sup>	V <sub>CC</sub> = Max V <sub>OUT</sub> = 5.5V V <sub>OUT</sub> = 0.45V V <sub>OUT</sub> = 0V		1 -1	40 -40 -70	μA mA
I <sub>CC</sub>	V <sub>CC</sub> supply current <sup>9</sup>	V <sub>CC</sub> = Max		120	180	mA
<b>Capacitance<sup>7</sup></b>						
C <sub>IN</sub> C <sub>OUT</sub>	Input Output	V <sub>CC</sub> = 5.0V V <sub>IN</sub> = 2.0V V <sub>OUT</sub> = 2.0V		8 10		pF

**NOTES:**

- Stresses above those listed under "Absolute Maximum Ratings" may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.
- All typical values are at V<sub>CC</sub> = 5V, T<sub>A</sub> = +25°C.
- All voltage values are with respect to network ground terminal.
- Test one at a time.
- Measured with V<sub>IL</sub> applied to  $\overline{\text{OE}}$  and a logic high stored, or with V<sub>IH</sub> applied to PR.
- Measured with a programmed logic condition for which the output is at a low logic level, and V<sub>IL</sub> applied to PR/ $\overline{\text{OE}}$ . Output sink current is supplied through a resistor to V<sub>CC</sub>.
- Measured with V<sub>IH</sub> applied to PR/ $\overline{\text{OE}}$ .
- Duration of short circuit should not exceed 1 second.
- I<sub>CC</sub> is measured with the PR/ $\overline{\text{OE}}$  input grounded, all other inputs at 4.5V and the outputs open.

**5**



## Field-Programmable Logic Sequencer (14 × 48 × 6)

PLS167A

**AC ELECTRICAL CHARACTERISTICS**  $R_1 = 470\Omega$ ,  $R_2 = 1k\Omega$ ,  $C_L = 30pF$ ,  $0^\circ C \leq T_A \leq +75^\circ C$ ,  $4.75V \leq V_{CC} \leq 5.25V$ 

SYMBOL	PARAMETER	TO	FROM	LIMITS			UNIT
				Min	Typ <sup>1</sup>	Max	
<b>Pulse width<sup>3</sup></b>							
t <sub>CKH</sub>	Clock <sup>2</sup> high	CK -	CK +	25	15		ns
t <sub>CKL</sub>	Clock low	CK +	CK -	25	15		
t <sub>CKP1B</sub>	Period (without Complement Array)	CK +	CK +	50	40		
t <sub>CKP2B</sub>	Period (with Complement Array)	CK +	CK +	80	50		
t <sub>PRH</sub>	Preset pulse	PR -	PR +	25	15		
<b>Setup time<sup>3</sup></b>							
t <sub>IS1A</sub>	Input	CK +	Input ±	40			ns
t <sub>IS1B</sub>	Input	CK +	Input ±	30			
t <sub>IS2A</sub>	Input (through Complement Array)	CK +	Input ±	70			
t <sub>IS2B</sub>	Input (through Complement Array)	CK +	Input	60			
t <sub>VS</sub>	Power-on preset	CK -	V <sub>CC</sub> +	0	-10		
t <sub>PRS</sub>	Preset	CK -	PR -	0	-10		
<b>Hold time</b>							
t <sub>IH</sub>	Input	Input ±	CK +	5	-10		ns
<b>Propagation delay</b>							
t <sub>CKO</sub>	Clock	Output ±	CK +		15	20	ns
t <sub>OE</sub>	Output enable	Output -	OE -		20	30	
t <sub>OD</sub>	Output disable	Output +	OE +		20	30	
t <sub>PR</sub>	Preset	Output +	PR +		18	30	
t <sub>PPR</sub>	Power-on preset	Output +	V <sub>CC</sub> +		0	10	
<b>Frequency of operation<sup>3</sup></b>							
f <sub>MaxB</sub>	Without Complement Array					20.0	MHz
f <sub>MaxB</sub>	With Complement Array					12.5	

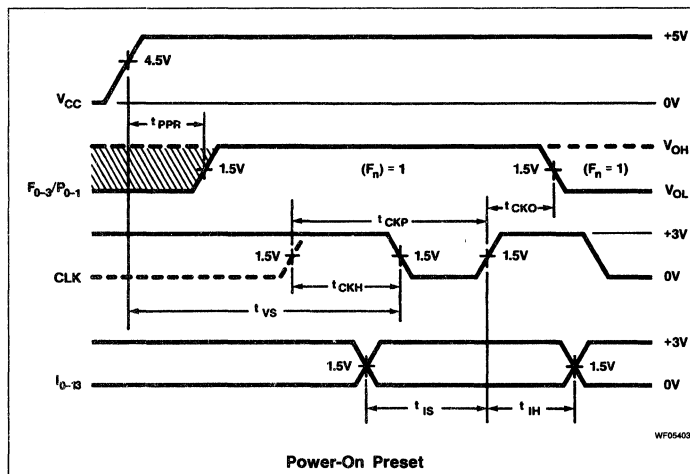
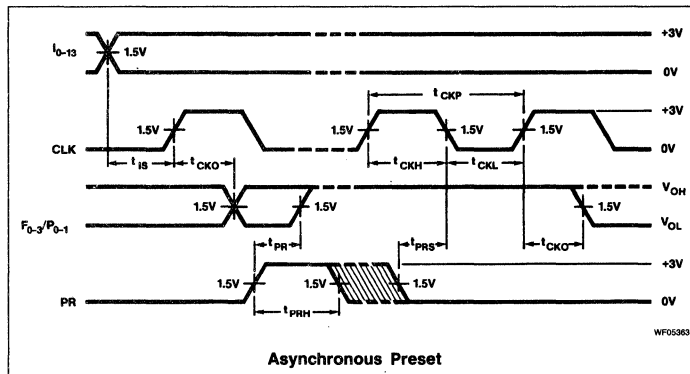
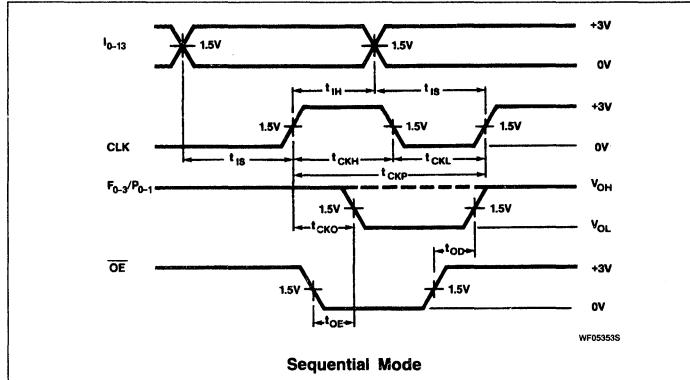
**NOTES:**

- All typical values are at  $V_{CC} = 5V$ ,  $T_A = +25^\circ C$ .
- To prevent spurious clocking, clock rise time (10% - 90%) < 30ns.
- See "Speed vs. OR Loading" diagrams.

# Field-Programmable Logic Sequencer (14 × 48 × 6)

PLS167A

## TIMING DIAGRAMS



## TIMING DEFINITIONS

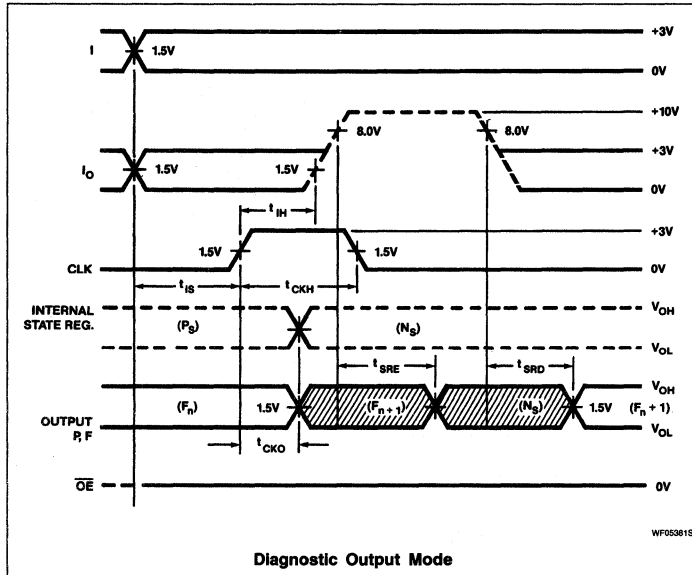
- $t_{CKH}$  Width of input clock pulse.
- $t_{CKL}$  Interval between clock pulses.
- $t_{CKP1}$  Clock period – when not using Complement array.
- $t_{IS1}$  Required delay between beginning of valid input and positive transition of clock.
- $t_{CKP2}$  Clock period – when using complement array.
- $t_{IS2}$  Required delay between beginning of valid input and positive transition of clock, when using optional Complement Array (two passes necessary through the AND array).
- $t_{VS}$  Required delay between  $V_{CC}$  (after power-on) and negative transition of clock preceding first reliable clock pulse.
- $t_{PRS}$  Required delay between negative transition of asynchronous Preset and negative transition of clock preceding first reliable clock pulse.
- $t_{IH}$  Required delay between positive transition of clock and end of valid input data.
- $t_{CKO}$  Delay between positive transition of clock and when Outputs become valid (with PR/OE Low).
- $t_{OE}$  Delay between beginning of Output Enable Low and when Outputs become valid.
- $t_{OD}$  Delay between beginning of Output Enable High and when Outputs are in the off state.
- $t_{SRE}$  Delay between input  $I_0$  transition to Diagnostic mode and when the Outputs reflect the contents of the State Register.
- $t_{SRD}$  Delay between input  $I_0$  transition to Logic mode and when the Outputs reflect the contents of the Output Register.
- $t_{PR}$  Delay between positive transition of Preset and when Outputs become valid at "1".
- $t_{PPR}$  Delay between  $V_{CC}$  (after power-on) and when Outputs become preset at "1".
- $t_{PRH}$  Width of preset input pulse.
- $f_{MAX}$  Maximum clock frequency.

5

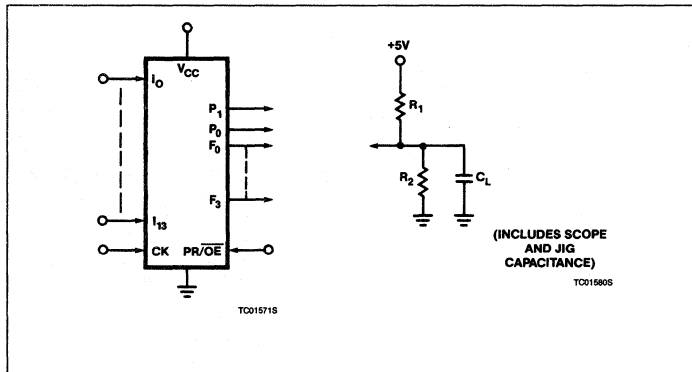
# Field-Programmable Logic Sequencer (14 × 48 × 6)

PLS167A

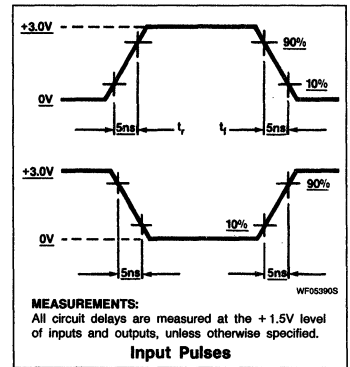
## TIMING DIAGRAMS (Continued)



## TEST LOAD CIRCUIT



## VOLTAGE WAVEFORM



**MEASUREMENTS:**  
All circuit delays are measured at the +1.5V level of inputs and outputs, unless otherwise specified.  
**Input Pulses**

Field-Programmable Logic Sequencer (14 × 48 × 6)

PLS167A

SPEED VS. "OR" LOADING

The maximum frequency at which the FPLS can be clocked while operating in *sequential mode* is given by:

$$(1/f_{MAX}) = t_{CY} = t_{IS} + t_{CKO}$$

This frequency depends on the number of transition terms  $T_n$  used. Having all 48 terms connected in the AND array does not appreciably impact performance; but the number of terms connected to each OR line affects  $t_{IS}$ , due to capacitive loading. The effect of this loading can be seen in Figure 1, showing the variation of  $t_{IS1}$  with the number of terms connected per OR.

The AC electrical characteristics contain two limits for the parameters  $t_{IS1}$  and  $t_{IS2}$ . The first,  $t_{IS1A}$  is guaranteed for a device with 24 terms connected to any OR line.  $t_{IS1B}$  is guaranteed for a device with 16 terms connected to any OR line.

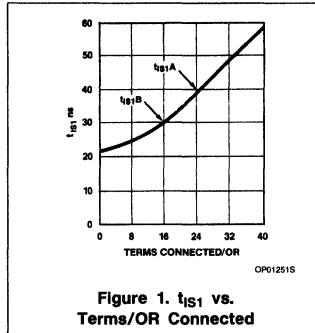


Figure 1.  $t_{IS1}$  vs. Terms/OR Connected

The three other entries in the AC table,  $t_{IS2}$  A and B are corresponding 24 and 16 term limits when using the on-chip Complement Array.

The worst case of  $t_{IS}$  for a given application can be determined by identifying the OR line with the maximum number of  $T_n$  connections. This can be done by referring to the interconnect pattern in the FPLS logic

diagram, typically illustrated in Figure 2, or by counting the maximum number of "H" or "L" entries in one of the columns of the device Program Table.

This number plotted on the curve in Figure 1 will yield the worst case  $t_{IS}$  and, by implication, the maximum clocking frequency for reliable operation.

*Note that for maximum speed all UNUSED transition terms should be disconnected from the OR array.*

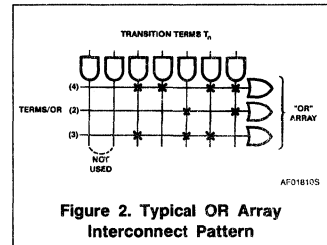


Figure 2. Typical OR Array Interconnect Pattern

# Field-Programmable Logic Sequencer (14 × 48 × 6)

PLS167A

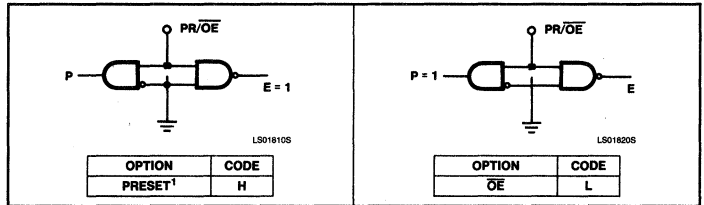
## LOGIC PROGRAMMING

The FPLS can be programmed by means of Logic programming equipment.

With Logic programming, the AND/OR gate input connections necessary to implement the desired logic function are coded directly from the State Diagram using the Program Table on the following page.

In this table, the logic state or action of control variables C, I, P, N, and F, associated with each Transition Term  $T_n$ , is assigned a symbol which results in the proper fusing pattern of corresponding link pairs, defined as follows:

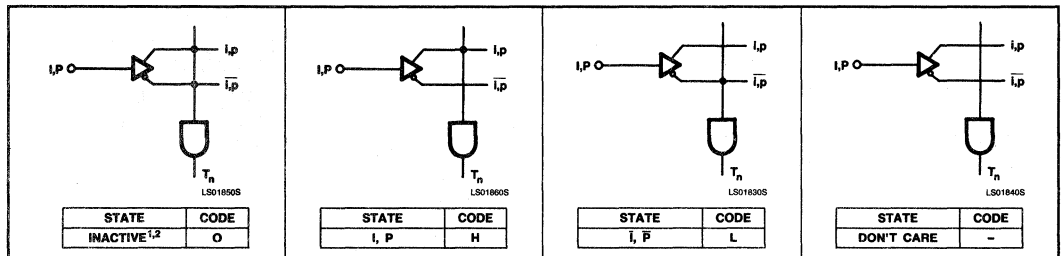
## PRESET/ $\overline{OE}$ OPTION - (P/E)



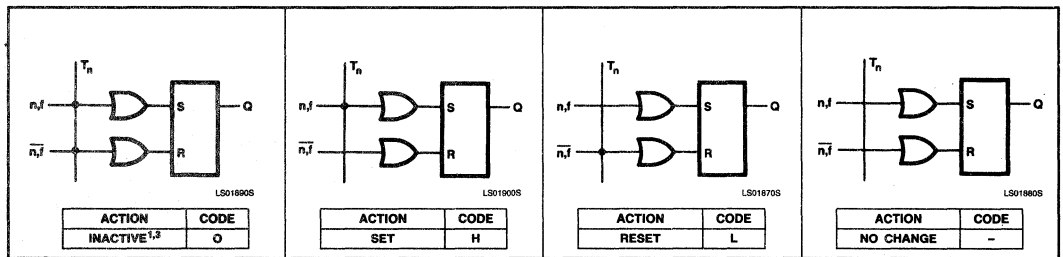
### PROGRAMMING:

The PLS167A has a power-up preset feature. This feature insures that the device will power-up in a known state with all register elements (state and output register) at a logic High (H). When programming the device it is important to realize this is the initial state of the device. You *must* provide a next state jump if you do not wish to use all Highs (H) as the present state.

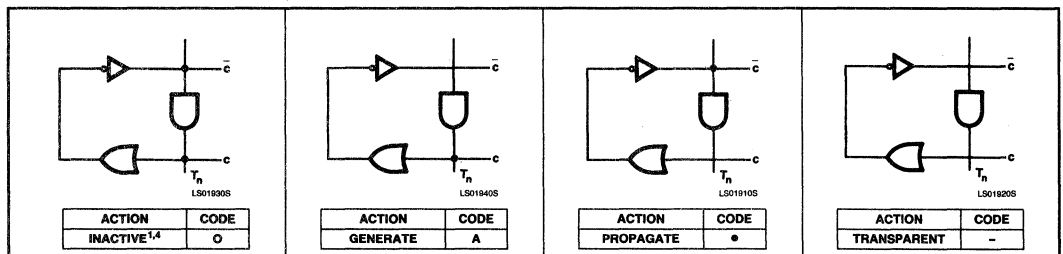
## "AND" ARRAY - (I), (P)



## "OR" ARRAY - (N), (F)



## "COMPLEMENT" ARRAY - (C)



### NOTES:

1. This is the initial unprogrammed state of all links.
2. Any gate  $T_n$  will be unconditionally inhibited if both the true and the complement of any input (I, P) are left intact.
3. To prevent simultaneous Set and Reset flip-flop commands, this state is not allowed for N and F link pairs coupled to active gates  $T_n$  (see flip-flop truth tables).
4. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates  $T_n$ .



# Field-Programmable Logic Sequencer (14 × 48 × 6)

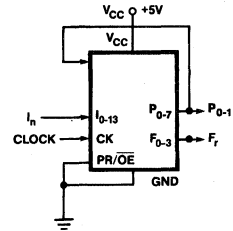
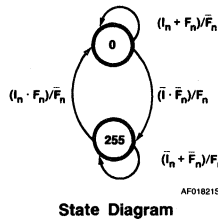
PLS167A

## TEST ARRAY

The FPLS may be subjected to AC and DC parametric tests prior to programming via an on-chip test array.

The array consists of test transition terms 48 and 49, factory programmed as shown below.

Testing is accomplished by clocking the FPLS and applying the proper input sequence to  $I_{0-13}$  as shown in the test circuit timing diagram.



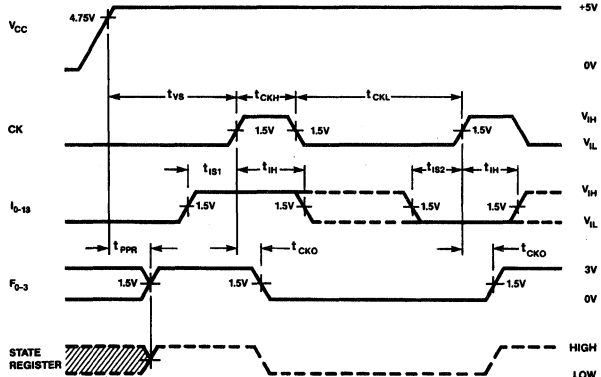
TERM	AND																						
	C	INPUT (In)							PRESENT STATE (Pa)														
		1 3	1 2	1 1	1 0	9	8	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
48	A	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
49	●	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L

OPTION (P/E)														H	
OR															
NEXT STATE (Ns)							OUTPUT (Fr)								
7	6	5	4	3	2	1	0	3	2	1	0	3	2	1	0
L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H

TB016005

Test Array Program

Both terms 48 and 49 must be deleted during user programming to avoid interfering with the desired logic function. This is accomplished automatically by any of Signetics' qualified programming equipment.



Test Circuit Timing Diagram

TERM	AND																						
	C	INPUT (Im)							PRESENT STATE (Ps)														
		1 3	1 2	1 1	1 0	9	8	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
48	-	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
49	●	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L

OPTION (P/E)														H	
OR															
NEXT STATE (Ns)							OUTPUT (Fr)								
7	6	5	4	3	2	1	0	3	2	1	0	3	2	1	0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

TB016105

Test Array Deleted

## PLS168 Field-Programmable Logic Sequencer (12 × 48 × 8)

Signetics Programmable Logic  
Product Specification

### Application Specific Products • Series 24

#### DESCRIPTION

The PLS168 is a bipolar, programmable state machine of the Mealy type. It contains logic AND-OR gate arrays with user programmable connections which control the inputs of on-chip State and Output Registers. These consist respectively of 10  $Q_p$ , and 4  $Q_f$  edge-triggered, clocked S/R flip-flops, with an Asynchronous Preset option.

All flip-flops are unconditionally preset to "1" during power turn-on.

The AND array combines 12 external inputs,  $I_{0-11}$ , with 10 internal inputs,  $P_{0-9}$ , fed back from the State Register to form up to 48 transition terms (AND terms). In addition,  $P_{0-3}$  of the internal State Register are brought off-chip to allow extending the Output Register to 8 bits, if so desired.

All transition terms can include True, False, or Don't Care states of the controlling variables, and are merged in the OR array to issue next-state and next-output commands to their respective registers on the Low-to-High transition of the Clock pulse.

Both True and Complement transition terms can be generated by optional use of the internal variable (C) from the Complement Array. Also, if desired, the Preset input can be converted to output-

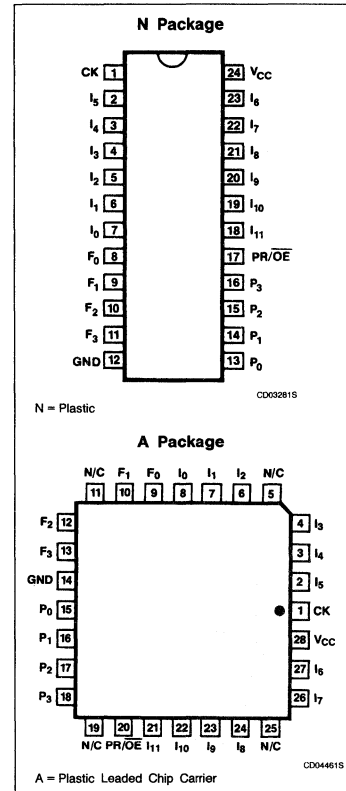
enable function, as an additional user programmable option.

Order codes for this device are contained on the pages following.

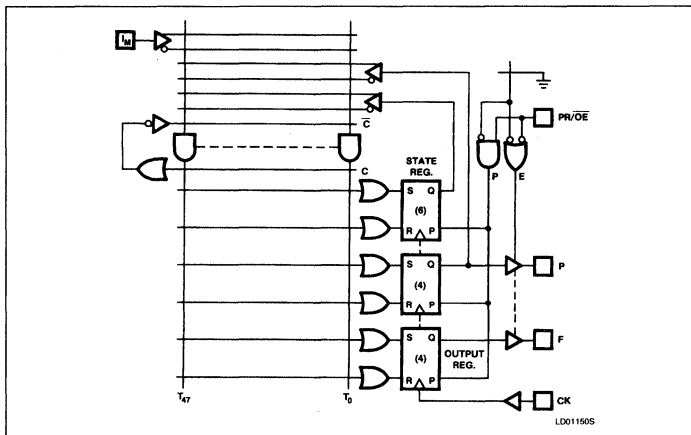
#### FEATURES

- Field-Programmable (Ni-Cr link)
- 12 True/Complement buffered inputs
- 48 programmable AND gates
- 29 programmable OR gates
- 10-bit State Register
- 4-bit shared State/Output Register
- 4-bit Output Register
- Transition Complement Array
- Programmable Asynchronous Preset/Output Enable
- Positive edge-triggered clock
- Power-on preset to logic "1" of all registers
- Automatic logic "HOLD" state via S/R flip-flops
- On-chip Test Array
- $f_{MAX}$ : 13.9MHz
- Power: 600mW (typ.)
- TTL compatible
- Tri-state outputs
- Single +5V supply
- 300mil-wide 24-pin DIP

#### PIN CONFIGURATIONS



#### FUNCTIONAL DIAGRAM



#### APPLICATIONS

- Interface protocols
- Sequence detectors
- Peripheral controllers
- Timing generators
- Sequential circuits
- Elevator controllers
- Security locking systems
- Counters
- Shift registers



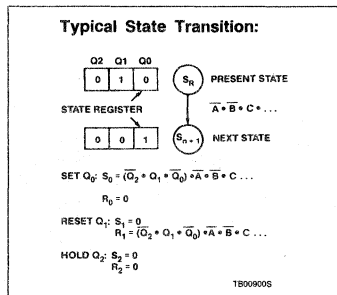
# Field-Programmable Logic Sequencer (12 × 48 × 8)

PLS168

## PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION	POLARITY
1	CK	<b>Clock:</b> The Clock input to the State and Output Registers. A Low-to-High transition on this line is necessary to update the contents of both registers.	Active-High
2-6 18-23	I <sub>1-11</sub>	<b>Logic Inputs:</b> The 11 external inputs to the AND array used to program jump conditions between machine states, as determined by a given logic sequence.	Active-High/Low
7	I <sub>0</sub>	<b>Logic/Diagnostic Input:</b> A 12th external logic input to the AND array, as above, when exercised with standard TTL levels. When I <sub>0</sub> is held at +10V, device outputs F <sub>2-3</sub> and P <sub>0-3</sub> reflect the contents of State Register bits P <sub>4-9</sub> (see Diagnostic Output Mode diagram). The contents of flip-flops P <sub>0-1</sub> and F <sub>0-3</sub> remain unaltered.	Active-High/Low
13-16	P <sub>0-3</sub>	<b>Logic/Diagnostic Outputs:</b> Four device outputs which normally reflect the contents of State Register bits P <sub>0-3</sub> . When I <sub>0</sub> is held at +10V these pins reflect (P <sub>6-P<sub>9</sub></sub> ).	Active-High
10-11	F <sub>2-F<sub>3</sub></sub>	<b>Logic/Diagnostic Outputs:</b> Two register bits (F <sub>2-F<sub>3</sub></sub> ) which normally reflect Output Register bits (Q <sub>2-Q<sub>3</sub></sub> ). When I <sub>0</sub> is held at +10V these pins reflect (P <sub>4-P<sub>5</sub></sub> ).	Active-High
17	PR/ $\overline{OE}$	<b>Preset or Output Enable Input:</b> A user programmable function: <ul style="list-style-type: none"> <li><b>Preset:</b> Provides an Asynchronous Preset to logic "1" of all State and Output Register bits. Preset overrides Clock, and when held high, clocking is inhibited and P<sub>0-9</sub> and F<sub>0-3</sub> are high. Normal clocking resumes with the first full clock pulse following a High-to-Low clock transition, after Preset goes low.</li> <li><b>Output Enable:</b> Provides an Output Enable function to all output buffers.</li> </ul>	Active-High (H)  Active-Low (L)
8, 9	F <sub>0-F<sub>1</sub></sub>	<b>Logic Output:</b> Two device outputs which reflect Output Registers Q <sub>0-Q<sub>1</sub></sub> . When I <sub>0</sub> is held at +10V F <sub>0-F<sub>1</sub></sub> = Logic "1".	

## LOGIC FUNCTION



## TRUTH TABLE 1, 2, 3, 4, 5, 6

V <sub>CC</sub>	OPTION		I <sub>0</sub>	CK	S	R	Q <sub>P/F</sub>	F	
	PR	$\overline{OE}$							
+5V	H		*	X	X	X	H	H	
	L		+10V	X	X	X	Q <sub>n</sub>	(Q <sub>P</sub> ) <sub>n</sub>	
	L		X	X	X	X	Q <sub>n</sub>	(Q <sub>F</sub> ) <sub>n</sub>	
		H	*	X	X	X	Q <sub>n</sub>	Hi-Z	
		L	+10V	X	X	X	Q <sub>n</sub>	(Q <sub>P</sub> ) <sub>n</sub>	
		L	X	X	X	X	Q <sub>n</sub>	(Q <sub>F</sub> ) <sub>n</sub>	
		L	X	↑	L	L	Q <sub>n</sub>	(Q <sub>F</sub> ) <sub>n</sub>	
		L	X	↑	L	H	L	L	
		L	X	↑	H	L	H	H	
		L	X	↑	H	H	IND.	IND.	
	↑	X	X	X	X	X	X	H	

### NOTES:

## VIRGIN STATE

A factory shipped virgin device contains all fusible links intact, such that:

- PR/ $\overline{OE}$  option is set to PR. Thus, all outputs will be at "1", as preset by initial power-up procedure.
- All transition terms are disabled (0).
- All S/R flip-flop inputs are disabled (0).
- The device can be clocked via a Test Array pre-programmed with a standard test pattern.

NOTE: The Test Array pattern MUST be deleted before incorporating a user program. This is accomplished automatically by any Signetics qualified programming equipment.

### 1. Positive Logic:

$$S/R = T_0 + T_1 + T_2 + \dots + T_{47}$$

$$T_n = C(I_0, I_1, I_2, \dots) (P_0, P_1, \dots, P_9)$$

2. Either Preset (Active-High) or Output Enable (Active-Low) are available, but not both. The desired function is a user programmable option.

3. ↑ denotes transition from Low to High level.

4. R = S = High is an illegal input condition.

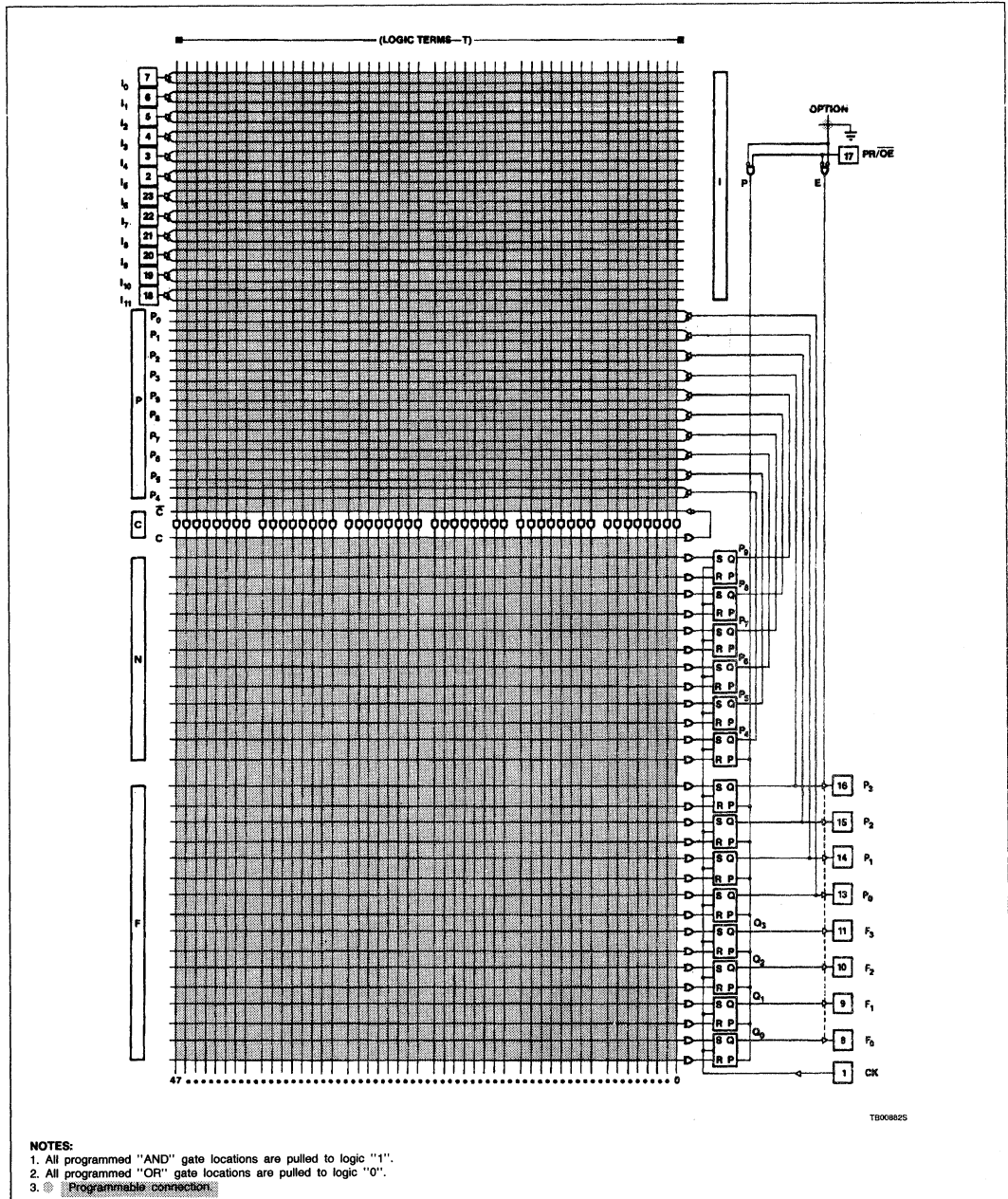
5. \* = H/L/+10V

6. X = Don't care (≤ 5.5V)

# Field-Programmable Logic Sequencer (12 × 48 × 8)


PLS168

## FPLS LOGIC DIAGRAM



5

**NOTES:**

1. All programmed "AND" gate locations are pulled to logic "1".
2. All programmed "OR" gate locations are pulled to logic "0".
3.  Programmable connection.

T80082S

## Field-Programmable Logic Sequencer (12 × 48 × 8)

PLS168

## ORDERING INFORMATION

DESCRIPTION	ORDER CODE
24-pin Plastic DIP 300mil-wide	PLS168N
28-pin Plastic Leaded Chip Carrier	PLS168A

## THERMAL RATING

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS<sup>1</sup>

SYMBOL	PARAMETER	RATING		UNIT
		Min	Max	
V <sub>CC</sub>	Supply voltage		+7	V <sub>DC</sub>
V <sub>IN</sub>	Input voltage		+5.5	V <sub>DC</sub>
V <sub>OUT</sub>	Output voltage		+5.5	V <sub>DC</sub>
I <sub>IN</sub>	Input currents	-30	+30	mA
I <sub>IN</sub>	Output currents		+100	mA
T <sub>A</sub> T <sub>STG</sub>	Temperature range Operating Storage	0 -65	+75 +150	°C

## Field-Programmable Logic Sequencer (12 × 48 × 8)

PLS168

DC ELECTRICAL CHARACTERISTICS  $0^{\circ}\text{C} \leq T_A \leq 75^{\circ}\text{C}$ ,  $4.75\text{V} \leq V_{CC} \leq 5.25\text{V}$ 

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ <sup>2</sup>	Max	
<b>Input voltage<sup>3</sup></b>						
V <sub>IH</sub> V <sub>IL</sub> V <sub>IC</sub>	High Low Clamp <sup>4</sup>	V <sub>CC</sub> = Max V <sub>CC</sub> = Min V <sub>CC</sub> = Min, I <sub>IN</sub> = -12mA	2	-0.8	0.8 -1.2	V
<b>Output voltage<sup>3</sup></b>						
V <sub>OH</sub> V <sub>OL</sub>	High <sup>5</sup> Low <sup>6</sup>	V <sub>CC</sub> = Min I <sub>OH</sub> = -2mA I <sub>OL</sub> = 9.6mA	2.4	0.35	0.45	V
<b>Input current</b>						
I <sub>IH</sub> I <sub>IL</sub> I <sub>IL</sub>	High Low Low (CK input)	V <sub>IN</sub> = 5.5V V <sub>IN</sub> = 0.45V V <sub>IN</sub> = 0.45V		< 1 -10 -50	25 -100 -250	μA
<b>Output current</b>						
I <sub>O(OFF)</sub> I <sub>OS</sub>	Hi-Z state <sup>7</sup> Short circuit <sup>4,8</sup>	V <sub>CC</sub> = Max V <sub>OUT</sub> = 5.5V V <sub>OUT</sub> = 0.45V V <sub>OUT</sub> = 0V	-15	1 -1	40 -40 -70	μA mA
I <sub>CC</sub>	V <sub>CC</sub> supply current <sup>9</sup>	V <sub>CC</sub> = Max		120	180	mA
<b>Capacitance<sup>7</sup></b>						
C <sub>IN</sub> C <sub>OUT</sub>	Input Output	V <sub>CC</sub> = 5.0V V <sub>IN</sub> = 2.0V V <sub>OUT</sub> = 2.0V		8 10		pF

## NOTES:

- Stresses above those listed under "Absolute Maximum Ratings" may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.
- All typical values are at V<sub>CC</sub> = 5V, T<sub>A</sub> = +25°C.
- All voltage values are with respect to network ground terminal.
- Test one at a time.
- Measured with V<sub>IL</sub> applied to  $\overline{\text{OE}}$  and a logic high stored, or with V<sub>IH</sub> applied to PR.
- Measured with a programmed logic condition for which the output is at a low logic level, and V<sub>IL</sub> applied to PR/ $\overline{\text{OE}}$ . Output sink current is supplied through a resistor to V<sub>CC</sub>.
- Measured with V<sub>IH</sub> applied to PR/ $\overline{\text{OE}}$ .
- Duration of short circuit should not exceed 1 second.
- I<sub>CC</sub> is measured with the PR/ $\overline{\text{OE}}$  input grounded, all other inputs at 4.5V and the outputs open.

## Field-Programmable Logic Sequencer (12 × 48 × 8)

PLS168

**AC ELECTRICAL CHARACTERISTICS**  $R_1 = 470\Omega$ ,  $R_2 = 1k\Omega$ ,  $C_L = 30pF$ ,  $0^\circ C \leq T_A \leq +75^\circ C$ ,  $4.75V \leq V_{CC} \leq 5.25V$ 

SYMBOL	PARAMETER	TO	FROM	LIMITS			UNIT
				Min	Typ <sup>1</sup>	Max	
<b>Pulse width<sup>3</sup></b>							
$t_{CKH}$	Clock <sup>2</sup> high	CK -	CK +	25	15		ns
$t_{CKL}$	Clock low	CK +	CK -	25	15		
$t_{CKP1B}$	Period (without Complement Array)	CK +	CK +	80	40		
$t_{CKP2B}$	Period (with Complement Array)	CK +	CK +	120	60		
$t_{PRH}$	Preset pulse	PR -	PR +	25	15		
<b>Setup time<sup>3</sup></b>							
$t_{IS1A}$	Input	CK +	Input $\pm$	60			ns
$t_{IS1B}$	Input	CK +	Input $\pm$	50			
$t_{IS1C}$	Input	CK +	Input $\pm$	42			
$t_{IS2A}$	Input (through Complement Array)	CK +	Input $\pm$	90			
$t_{IS2B}$	Input (through Complement Array)	CK +	Input	80			
$t_{IS2C}$	Input (through Complement Array)	CK +	Input	72			
$t_{VS}$	Power-on preset	CK -	$V_{CC} +$	0	-10		
$t_{PRS}$	Preset	CK -	PR -	0	-10		
<b>Hold time</b>							
$t_{IH}$	Input	Input $\pm$	CK +	5	-10		ns
<b>Propagation delay</b>							
$t_{CKO}$	Clock	Output $\pm$	CK +		15	30	ns
$t_{OE}$	Output enable	Output -	OE -		20	30	
$t_{OD}$	Output disable	Output +	OE +		20	30	
$t_{PR}$	Preset	Output +	PR +		18	30	
$t_{PPR}$	Power-on preset	Output +	$V_{CC} +$		0	10	
<b>Frequency of operation<sup>3</sup></b>							
$f_{MAXC}$	Without Complement Array					13.9	MHz
$f_{MAXC}$	With Complement Array					9.8	

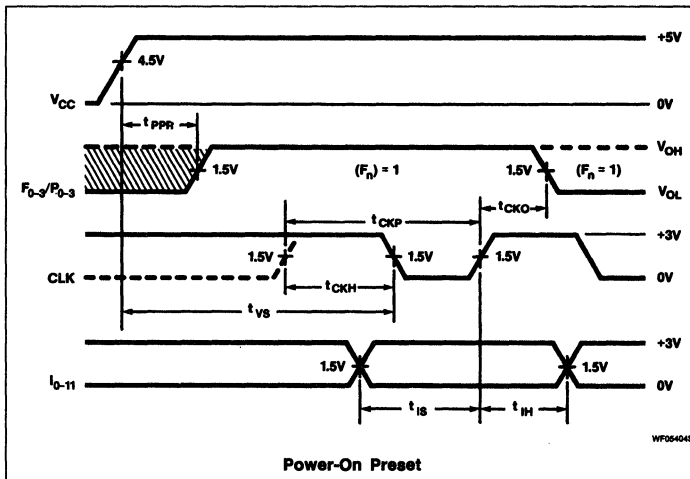
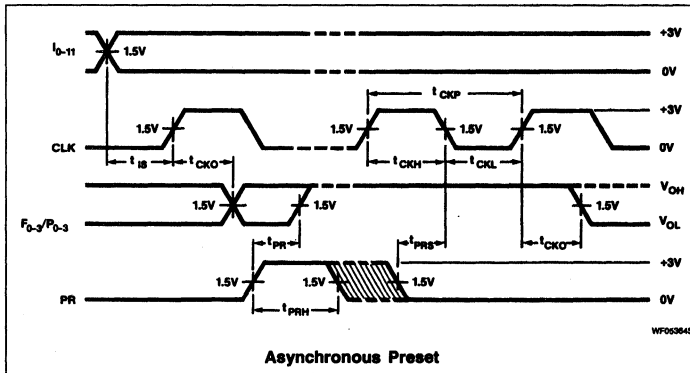
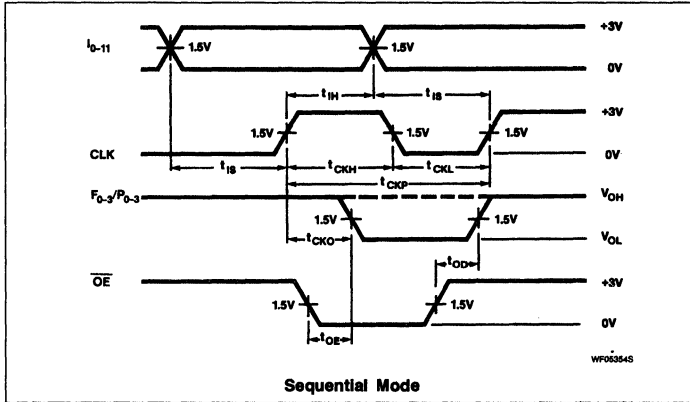
**NOTES:**

- All typical values are at  $V_{CC} = 5V$ ,  $T_A = +25^\circ C$ .
- To prevent spurious clocking, clock rise time (10% - 90%)  $\leq 30ns$ .
- See "Speed vs. OR Loading" diagrams.

# Field-Programmable Logic Sequencer (12 × 48 × 8)

PLS168

## TIMING DIAGRAMS



## TIMING DEFINITIONS

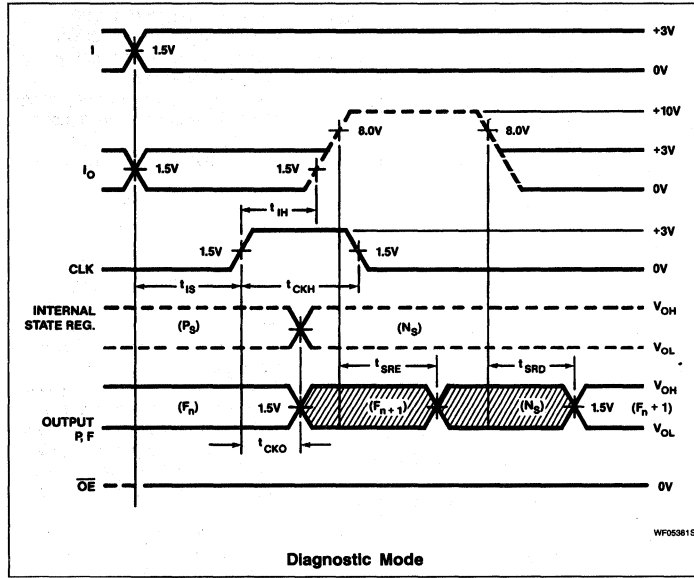
- $t_{CKH}$  Width of input clock pulse.
- $t_{CKL}$  Interval between clock pulses.
- $t_{CKP1}$  Clock period – when not using Complement array.
- $t_{IS1}$  Required delay between beginning of valid input and positive transition of clock.
- $t_{CKP2}$  Clock period – when using complement array.
- $t_{IS2}$  Required delay between beginning of valid input and positive transition of clock, when using optional Complement Array (two passes necessary through the AND array).
- $t_{VS}$  Required delay between  $V_{CC}$  (after power-on) and negative transition of clock preceding first reliable clock pulse.
- $t_{PRS}$  Required delay between negative transition of asynchronous Preset and negative transition of clock preceding first reliable clock pulse.
- $t_{IH}$  Required delay between positive transition of clock and end of valid input data.
- $t_{CKO}$  Delay between positive transition of clock and when Outputs become valid (with PR/OE Low).
- $t_{OE}$  Delay between beginning of Output Enable Low and when Outputs become valid.
- $t_{OD}$  Delay between beginning of Output Enable High and when Outputs are in the off state.
- $t_{SRE}$  Delay between input  $I_0$  transition to Diagnostic mode and when the Outputs reflect the contents of the State Register.
- $t_{SRD}$  Delay between input  $I_0$  transition to Logic mode and when the Outputs reflect the contents of the Output Register.
- $t_{PR}$  Delay between positive transition of Preset and when Outputs become valid "1".
- $t_{PPR}$  Delay between  $V_{CC}$  (after power-on) and when Outputs become preset at "1".
- $t_{PRH}$  Width of preset input pulse.
- $f_{MAX}$  Maximum clock frequency.

5

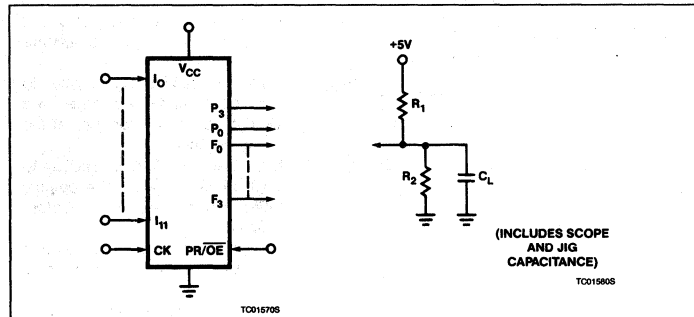
# Field-Programmable Logic Sequencer (12 × 48 × 8)

PLS168

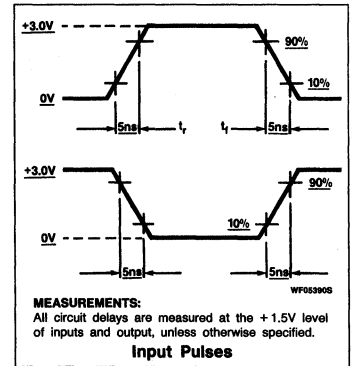
## TIMING DIAGRAMS (Continued)



## TEST LOAD CIRCUIT



## VOLTAGE WAVEFORM



# Field-Programmable Logic Sequencer (12 × 48 × 8)

PLS168

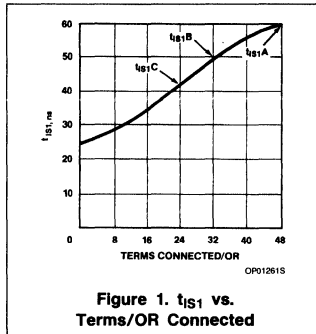
## SPEED VS. "OR" LOADING

The maximum frequency at which the FPLS can be clocked while operating in *sequential mode* is given by:

$$(1/f_{MAX}) = t_{CY} = t_{IS} + t_{CKO}$$

This frequency depends on the number of transition terms  $T_n$  used. Having all 48 terms connected in the AND array does not appreciably impact performance; but the number of terms connected to each OR line affects  $t_{IS}$ , due to capacitive loading. The effect of this loading can be seen in Figure 1, showing the variation of  $t_{IS1}$  with the number of terms connected per OR.

The AC electrical characteristics contain three limits for the parameters  $t_{IS1}$  and  $t_{IS2}$ . The first,  $t_{IS1A}$  is guaranteed for a device with 48 terms connected to any OR line.  $t_{IS1B}$  is guaranteed for a device with 32 terms connected to any OR line. And  $t_{IS1C}$  is guaranteed for a device with 24 terms connected to any OR line.



**Figure 1.  $t_{IS1}$  vs. Terms/OR Connected**

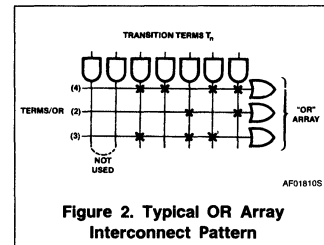
The three other entries in the AC table,  $t_{IS2}$  A, B and C are corresponding 48, 32 and 24 term limits when using the on-chip Complement Array.

The worst case of  $t_{IS}$  for a given application can be determined by identifying the OR line with the maximum number of  $T_n$  connections. This can be done by referring to the interconnect pattern in the FPLS logic

diagram, typically illustrated in Figure 2, or by counting the maximum number of "H" or "L" entries in one of the columns of the device Program Table.

This number plotted on the curve in Figure 1 will yield the worst case  $t_{IS}$  and, by implication, the maximum clocking frequency for reliable operation.

*Note that for maximum speed all UNUSED transition terms should be disconnected from the OR array.*



**Figure 2. Typical OR Array Interconnect Pattern**



# Field-Programmable Logic Sequencer (12 × 48 × 8)

PLS168

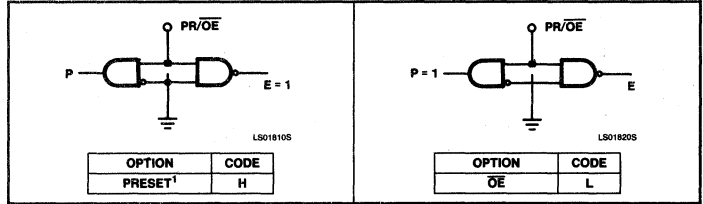
## LOGIC PROGRAMMING

The FPLS can be programmed by means of Logic programming equipment.

With Logic programming, the AND/OR gate input connections necessary to implement the desired logic function are coded directly from the State Diagram using the Program Table on the following page.

In this table, the logic state or action of control variables C, I, P, N, and F, associated with each Transition Term  $T_n$ , is assigned a symbol which results in the proper fusing pattern of corresponding links defined as follows:

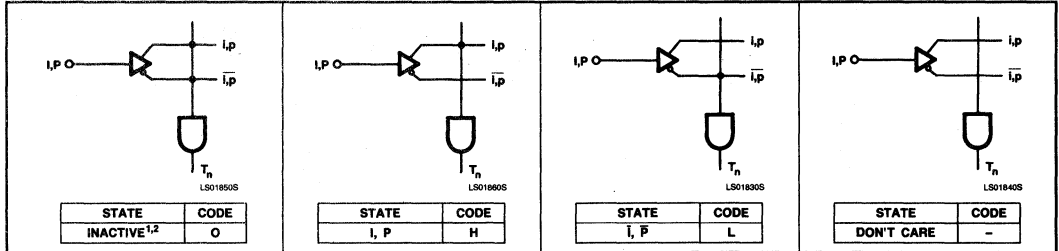
## PRESET/ $\overline{OE}$ OPTION - (P/E)



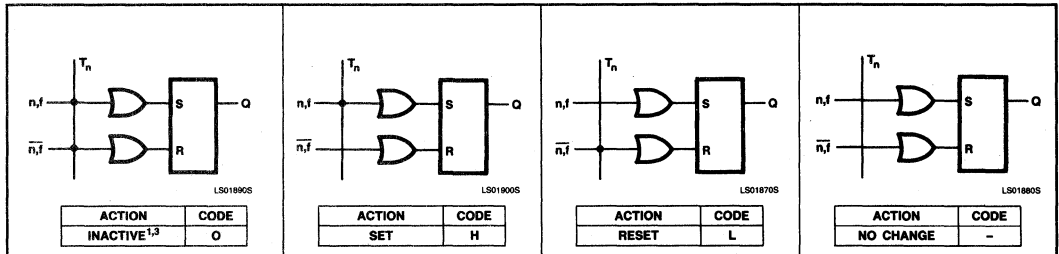
### PROGRAMMING:

The PLS168 has a power-up preset feature. This feature insures that the device will power-up in a known state with all register elements (state and output register) at a logic High (H). When programming the device it is important to realize this is the initial state of the device. You *must* provide a next state jump if you do not wish to use all Highs (H) as the present state.

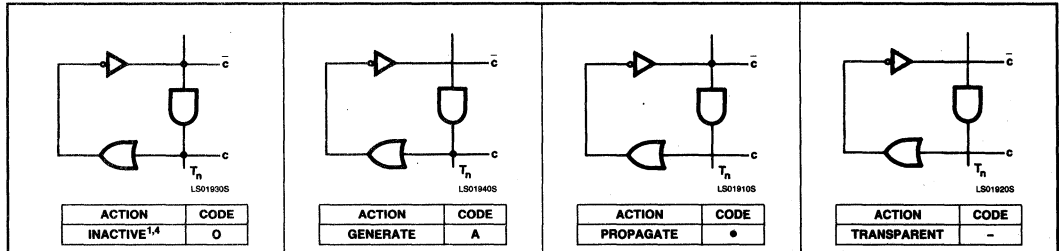
## "AND" ARRAY - (I), (P)



## "OR" ARRAY - (N), (F)



## "COMPLEMENT" ARRAY - (C)



### NOTES:

1. This is the initial unprogrammed state of all links.
2. Any gate  $T_n$  will be unconditionally inhibited if both the true and complement of any input (I or P) are left intact.
3. To prevent simultaneous Set and Reset flip-flop commands, this state is not allowed for N and F link pairs coupled to active gates  $T_n$  (see flip-flop truth tables).
4. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates  $T_n$ .



# Field-Programmable Logic Sequencer (12 × 48 × 8)

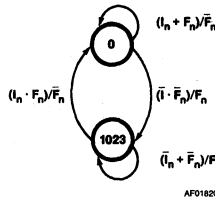
PLS168

## TEST ARRAY

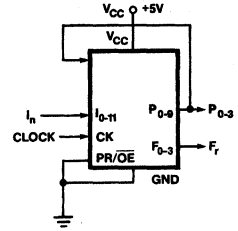
The FPLS may be subjected to AC and DC parametric tests prior to programming via an on-chip test array.

The array consists of test transition terms 48 and 49, factory programmed as shown below.

Testing is accomplished by clocking the FPLS and applying the proper input sequence to I<sub>0-11</sub> as shown in the test circuit timing diagram.



State Diagram

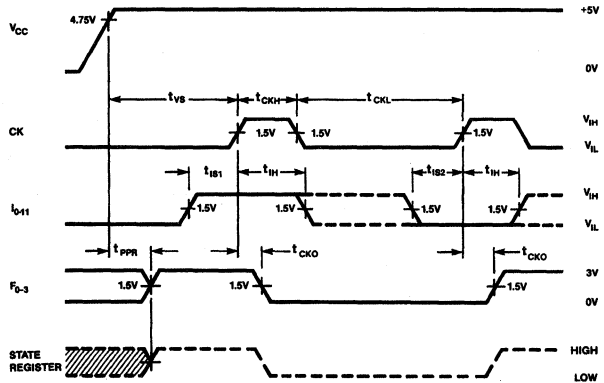


FPLS Under Test

AND													OPTION (P/E)													H																						
													OR																																			
T E R M	C <sub>n</sub>	INPUT (I <sub>m</sub> )											PRESENT STATE (P <sub>s</sub> )											REMARKS	NEXT STATE (N <sub>s</sub> )											OUTPUT (F <sub>r</sub> )												
		1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0		9	8	7	6	5	4	3	2	1	0	3	2	1	0										
48	A	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H		L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
49	•	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L		H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H

Test Array Program

Both terms 48 and 49 must be deleted during user programming to avoid interfering with the desired logic function. This is accomplished automatically by any of Signetics' qualified programming equipment.



Test Circuit Timing Diagram

AND													OPTION (P/E)													H																						
													OR																																			
T E R M	C <sub>n</sub>	INPUT (I <sub>m</sub> )											PRESENT STATE (P <sub>s</sub> )											REMARKS	NEXT STATE (N <sub>s</sub> )											OUTPUT (F <sub>r</sub> )												
		1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0		9	8	7	6	5	4	3	2	1	0	3	2	1	0										
48	~	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H		~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
49	•	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L		~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~

Test Array Deleted

## PLS168A

### Field-Programmable Logic Sequencer (12 × 48 × 8)

Signetics Programmable Logic  
Product Specification

#### Application Specific Products

##### • Series 24

#### DESCRIPTION

The PLS168A is a bipolar, programmable state machine of the Mealy type. It contains logic AND-OR gate arrays with user programmable connections which control the inputs of on-chip State and Output Registers. These consist respectively of 10  $Q_p$ , and 4  $Q_f$  edge-triggered, clocked S/R flip-flops, with an Asynchronous Preset option.

All flip-flops are unconditionally preset to "1" during power turn-on.

The AND array combines 12 external inputs,  $I_{0-11}$ , with 10 internal inputs,  $P_{0-9}$ , fed back from the State Register to form up to 48 transition terms (AND terms). In addition,  $P_{0-3}$  of the internal State Register are brought off-chip to allow extending the Output Register to 8 bits, if so desired.

All transition terms can include True, False, or Don't Care states of the controlling variables, and are merged in the OR array to issue next-state and next-output commands to their respective registers on the Low-to-High transition of the Clock pulse.

Both True and Complement transition terms can be generated by optional use of the internal variable (C) from the Complement Array. Also, if desired, the Preset input can be converted to output-

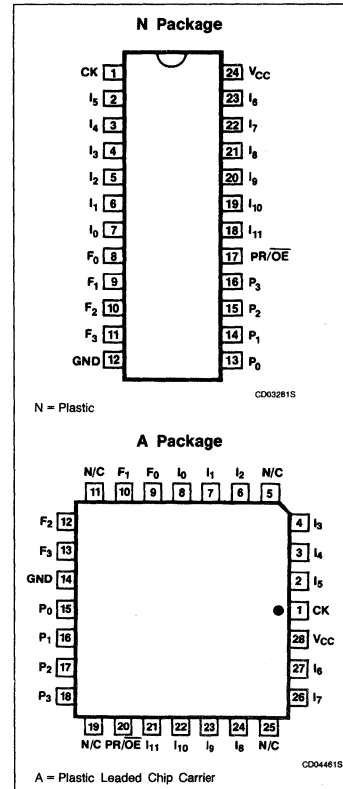
enable function, as an additional user programmable option.

Order codes for this device are contained on the pages following.

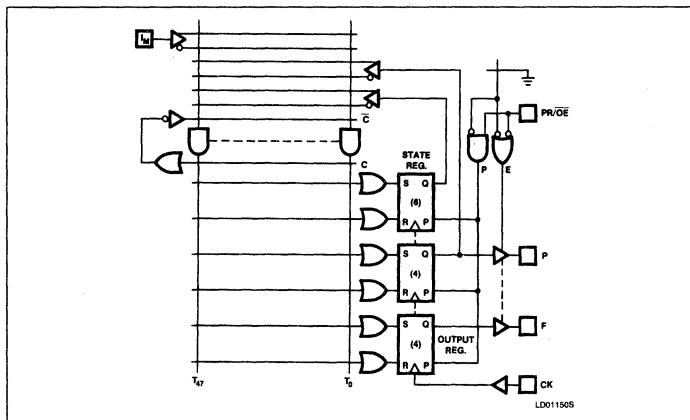
#### FEATURES

- Field-Programmable (Ni-Cr link)
- 12 True/Complement buffered inputs
- 48 programmable AND gates
- 29 programmable OR gates
- 10-bit State Register
- 4-bit shared State/Output Register
- 4-bit Output Register
- Transition Complement Array
- Programmable Asynchronous Preset/Output Enable
- Positive edge-triggered clock
- Power-on preset to logic "1" of all registers
- Automatic logic "HOLD" state via S/R flip-flops
- On-chip Test Array
- $f_{MAX}$ : 20MHz
- Power: 600mW (typ.)
- TTL compatible
- Tri-state outputs
- Single +5V supply
- 300mil-wide 24-pin DIP

#### PIN CONFIGURATIONS



#### FUNCTIONAL DIAGRAM



#### APPLICATIONS

- Interface protocols
- Sequence detectors
- Peripheral controllers
- Timing generators
- Sequential circuits
- Elevator controllers
- Security locking systems
- Counters
- Shift registers

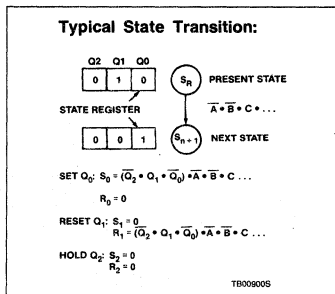
# Field-Programmable Logic Sequencer (12 × 48 × 8)

PLS168A

## PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION	POLARITY
1	CK	<b>Clock:</b> The Clock input to the State and Output Registers. A Low-to-High transition on this line is necessary to update the contents of both registers.	Active-High
2-6 18-23	I <sub>1-11</sub>	<b>Logic Inputs:</b> The 11 external inputs to the AND array used to program jump conditions between machine states, as determined by a given logic sequence.	Active-High/Low
7	I <sub>0</sub>	<b>Logic/Diagnostic Input:</b> A 12th external logic input to the AND array, as above, when exercised with standard TTL levels. When I <sub>0</sub> is held at +10V, device outputs F <sub>2-F3</sub> and P <sub>0-P3</sub> reflect the contents of State Register bits P <sub>4-9</sub> (see Diagnostic Output Mode diagram). The contents of flip-flops P <sub>0-1</sub> and F <sub>0-3</sub> remain unaltered.	Active-High/Low
13-16	P <sub>0-3</sub>	<b>Logic/Diagnostic Outputs:</b> Four device outputs which normally reflect the contents of State Register bits P <sub>0-3</sub> . When I <sub>0</sub> is held at +10V these pins reflect (P <sub>6-P9</sub> ).	Active-High
10-11	F <sub>2-F3</sub>	<b>Logic/Diagnostic Outputs:</b> Two register bits (F <sub>2-F3</sub> ) which reflect Output Register bits (Q <sub>2-Q3</sub> ). When I <sub>0</sub> is held at +10V these pins reflect (P <sub>4-P5</sub> ).	Active-High
17	PR/ $\overline{OE}$	<b>Preset or Output Enable Input:</b> A user programmable function: <ul style="list-style-type: none"> <li><b>Preset:</b> Provides an Asynchronous Preset to logic "1" of all State and Output Register bits. Preset overrides Clock, and when held high, clocking is inhibited and P<sub>0-9</sub> and F<sub>0-3</sub> are high. Normal clocking resumes with the first full clock pulse following a High-to-Low clock transition, after Preset goes low.</li> <li><b>Output Enable:</b> Provides an Output Enable function to all output buffers.</li> </ul>	Active-High (H)  Active-Low (L)
8, 9	F <sub>0-F1</sub>	<b>Logic Output:</b> Two device outputs which reflect Output Registers Q <sub>0-Q1</sub> . When I <sub>0</sub> is held at +10V F <sub>0-F1</sub> = Logic "1"	

## LOGIC FUNCTION



## VIRGIN STATE

A factory shipped virgin device contains all fusible links intact, such that:

- PR/ $\overline{OE}$  option is set to PR. Thus, all outputs will be at "1", as preset by initial power-up procedure.
- All transition terms are disabled (0).
- All S/R flip-flop inputs are disabled (0).
- The device can be clocked via a Test Array pre-programmed with a standard test pattern.

NOTE: The Test Array pattern MUST be deleted before incorporating a user program.

## TRUTH TABLE<sup>1, 2, 3, 4, 5, 6</sup>

V <sub>CC</sub>	OPTION		I <sub>0</sub>	CK	S	R	Q <sub>P/F</sub>	F	
	PR	$\overline{OE}$							
+5V	H	X	*	X	X	X	H	H	
	L	X	+10V	X	X	X	Q <sub>n</sub>	(Q <sub>P</sub> ) <sub>n</sub>	
	L	X	X	X	X	X	Q <sub>n</sub>	(Q <sub>F</sub> ) <sub>n</sub>	
	H	X	*	X	X	X	Q <sub>n</sub>	Hi-Z	
	L	X	+10V	X	X	X	Q <sub>n</sub>	(Q <sub>P</sub> ) <sub>n</sub>	
	L	X	X	X	X	X	Q <sub>n</sub>	(Q <sub>F</sub> ) <sub>n</sub>	
+5V	X	X	L	X	↑	L	L	Q <sub>n</sub>	(Q <sub>F</sub> ) <sub>n</sub>
			L	X	↑	H	L	L	L
			L	X	↑	H	L	H	H
			L	X	↑	H	H	IND.	IND.
↑	X	X	X	X	X	X	H		

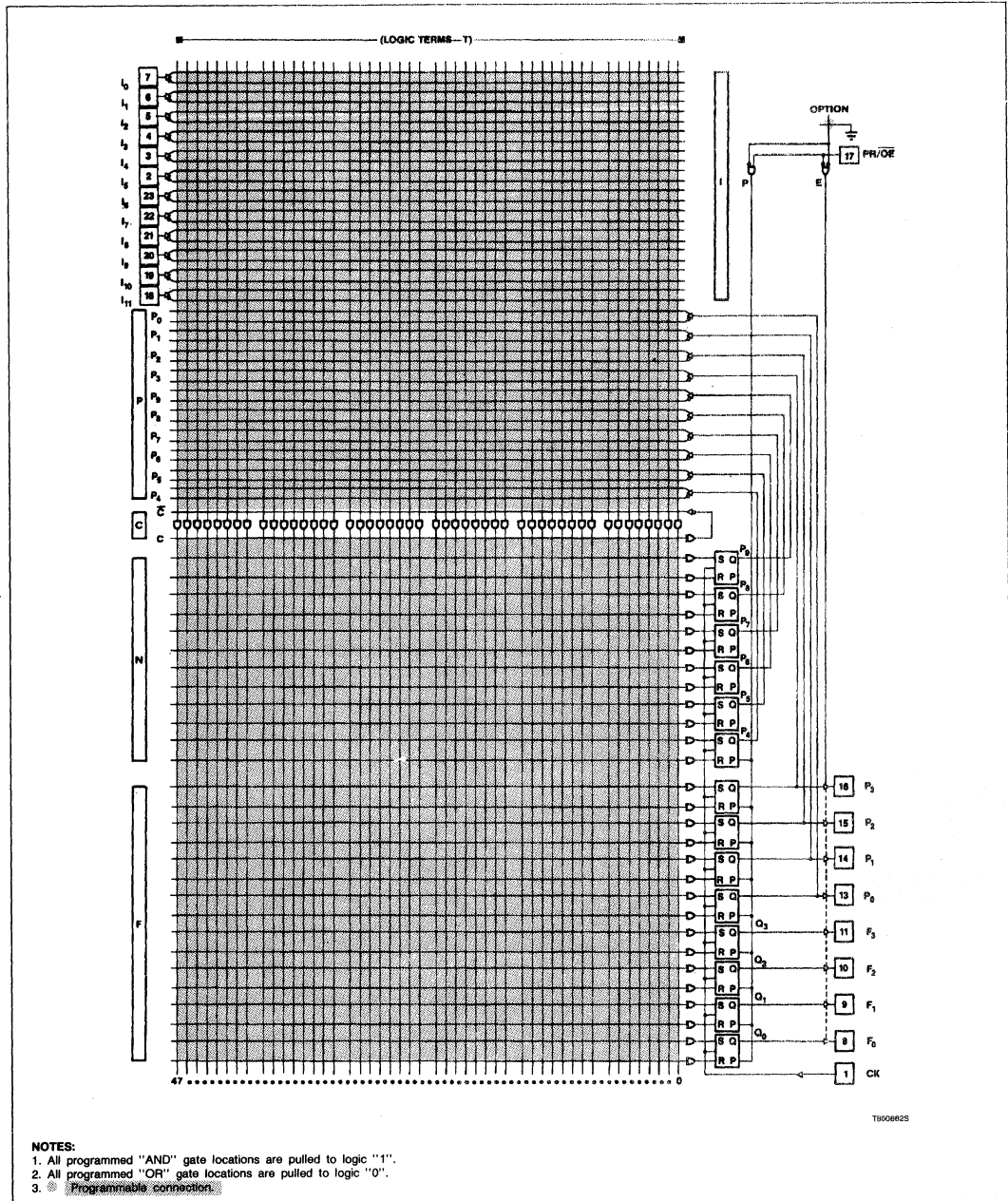
### NOTES:

- Positive Logic:  
S/R = T<sub>0</sub> + T<sub>1</sub> + T<sub>2</sub> + ... + T<sub>17</sub>  
T<sub>n</sub> = C(I<sub>0</sub> I<sub>1</sub> I<sub>2</sub>...) (P<sub>0</sub> P<sub>1</sub>...P<sub>9</sub>)
- Either Preset (Active-High) or Output Enable (Active-Low) are available, but not both. The desired function is a user programmable option.
- ↑ denotes transition from Low to High level.
- R = S = High is an illegal input condition.
- \* = H/L/+10V
- X = Don't Care (<= 5.5V)

# Field-Programmable Logic Sequencer (12 × 48 × 8)

PLS168A

## FPLS LOGIC DIAGRAM



## Field-Programmable Logic Sequencer (12 × 48 × 8)

PLS168A

## ORDERING INFORMATION

DESCRIPTION	ORDER CODE
24-pin Plastic DIP 300mil-wide	PLS168AN
28-pin Plastic Leaded Chip Carrier	PLS168AA

## THERMAL RATING

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS<sup>1</sup>

SYMBOL	PARAMETER	RATING		UNIT
		Min	Max	
V <sub>CC</sub>	Supply voltage		+7	V <sub>DC</sub>
V <sub>IN</sub>	Input voltage		+5.5	V <sub>DC</sub>
V <sub>OUT</sub>	Output voltage		+5.5	V <sub>DC</sub>
I <sub>IN</sub>	Input currents	-30	+30	mA
I <sub>IN</sub>	Output currents		+100	mA
T <sub>A</sub> T <sub>STG</sub>	Temperature range Operating Storage	0 -65	+75 +150	°C

## Field-Programmable Logic Sequencer (12 × 48 × 8)

PLS168A

DC ELECTRICAL CHARACTERISTICS  $0^{\circ}\text{C} \leq T_A \leq 75^{\circ}\text{C}$ ,  $4.75\text{V} \leq V_{CC} \leq 5.25\text{V}$ 

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ <sup>2</sup>	Max	
<b>Input voltage<sup>3</sup></b>						
V <sub>IH</sub> V <sub>IL</sub> V <sub>IC</sub>	High Low Clamp <sup>4</sup>	V <sub>CC</sub> = Max V <sub>CC</sub> = Min V <sub>CC</sub> = Min, I <sub>IN</sub> = -12mA	2		0.8 -1.2	V
<b>Output voltage<sup>3</sup></b>						
V <sub>OH</sub> V <sub>OL</sub>	High <sup>5</sup> Low <sup>6</sup>	V <sub>CC</sub> = Min I <sub>OH</sub> = -2mA I <sub>OL</sub> = 9.6mA	2.4	0.35	0.45	V
<b>Input current</b>						
I <sub>IH</sub> I <sub>IL</sub> I <sub>IL</sub>	High Low Low (CK input)	V <sub>IN</sub> = 5.5V V <sub>IN</sub> = 0.45V V <sub>IN</sub> = 0.45V		< 1 -10 -50	25 -100 -250	μA
<b>Output current</b>						
I <sub>O(OFF)</sub> I <sub>OS</sub>	Hi-Z state <sup>7</sup> Short circuit <sup>4,8</sup>	V <sub>CC</sub> = Max V <sub>OUT</sub> = 5.5V V <sub>OUT</sub> = 0.45V V <sub>OUT</sub> = 0V		1 -1	40 -40 -70	μA mA
I <sub>CC</sub>	V <sub>CC</sub> supply current <sup>9</sup>	V <sub>CC</sub> = Max		120	180	mA
<b>Capacitance<sup>7</sup></b>						
C <sub>IN</sub> C <sub>OUT</sub>	Input Output	V <sub>CC</sub> = 5.0V V <sub>IN</sub> = 2.0V V <sub>OUT</sub> = 2.0V		8 10		pF

## NOTES:

- Stresses above those listed under "Absolute Maximum Ratings" may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.
- All typical values are at V<sub>CC</sub> = 5V, T<sub>A</sub> = +25°C.
- All voltage values are with respect to network ground terminal.
- Test one at a time.
- Measured with V<sub>IL</sub> applied to  $\overline{\text{OE}}$  and a logic high stored, or with V<sub>IH</sub> applied to PR.
- Measured with a programmed logic condition for which the output is at a low logic level, and V<sub>IL</sub> applied to PR/ $\overline{\text{OE}}$ . Output sink current is supplied through a resistor to V<sub>CC</sub>.
- Measured with V<sub>IH</sub> applied to PR/ $\overline{\text{OE}}$ .
- Duration of short circuit should not exceed 1 second.
- I<sub>CC</sub> is measured with the PR/ $\overline{\text{OE}}$  input grounded, all other inputs at 4.5V and the outputs open.



## Field-Programmable Logic Sequencer (12 × 48 × 8)

PLS168A

AC ELECTRICAL CHARACTERISTICS  $R_1 = 470\Omega$ ,  $R_2 = 1k\Omega$ ,  $C_L = 30pF$ ,  $0^\circ C \leq T_A \leq +75^\circ C$ ,  $4.75V \leq V_{CC} \leq 5.25V$ 

SYMBOL	PARAMETER	TO	FROM	LIMITS			UNIT
				Min	Typ <sup>1</sup>	Max	
<b>Pulse width<sup>3</sup></b>							
$t_{CKH}$	Clock <sup>2</sup> high	CK -	CK +	25	15		ns
$t_{CKL}$	Clock low	CK +	CK -	25	15		
$t_{CKP1B}$	Period (without Complement Array)	CK +	CK +	50	40		
$t_{CKP2B}$	Period (with Complement Array)	CK +	CK +	80	50		
$t_{PRH}$	Preset pulse	PR -	PR +	25	15		
<b>Setup time<sup>3</sup></b>							
$t_{S1A}$	Input	CK +	Input $\pm$	40			ns
$t_{S1B}$	Input	CK +	Input $\pm$	30			
$t_{S2A}$	Input (through Complement Array)	CK +	Input $\pm$	70			
$t_{S2B}$	Input (through Complement Array)	CK +	Input	60			
$t_{VS}$	Power-on preset	CK -	$V_{CC} +$	0	-10		
$t_{PRS}$	Preset	CK -	PR -	0	-10		
<b>Hold time</b>							
$t_H$	Input	Input $\pm$	CK +	5	-10		ns
<b>Propagation delay</b>							
$t_{CKO}$	Clock	Output $\pm$	CK +		15	20	ns
$t_{OE}$	Output enable	Output -	OE -		20	30	
$t_{OD}$	Output disable	Output +	OE +		20	30	
$t_{PR}$	Preset	Output +	PR +		18	30	
$t_{PPR}$	Power-on preset	Output +	$V_{CC} +$		0	10	
<b>Frequency of operation<sup>3</sup></b>							
$f_{MAXB}$	Without Complement Array					20	MHz
$f_{MAXB}$	With Complement Array					12.5	

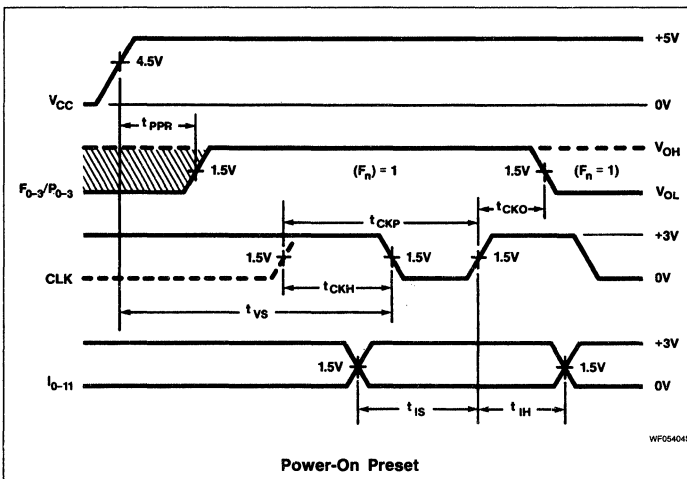
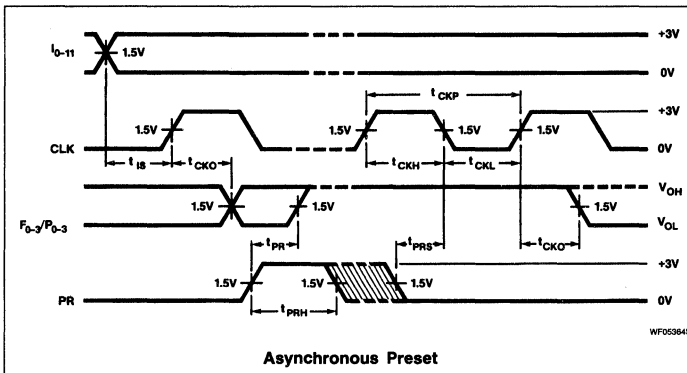
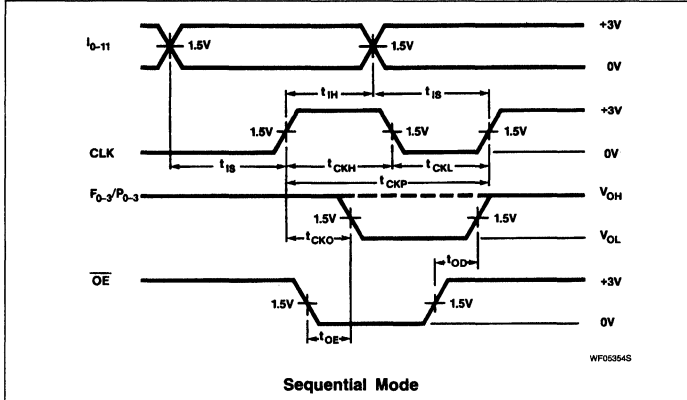
## NOTES:

- All typical values are at  $V_{CC} = 5V$ ,  $T_A = +25^\circ C$ .
- To prevent spurious clocking, clock rise time (10% - 90%)  $\leq 30ns$ .
- See "Speed vs. OR Loading" diagrams.

# Field-Programmable Logic Sequencer (12 × 48 × 8)

PLS168A

## TIMING DIAGRAMS



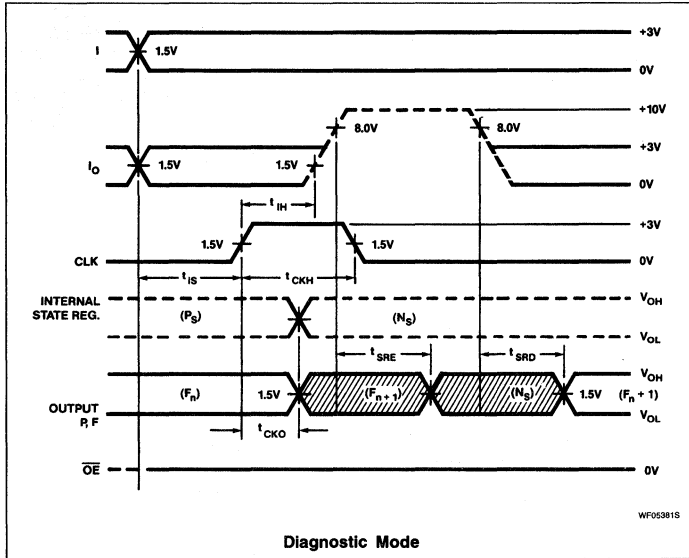
## TIMING DEFINITIONS

- $t_{CKH}$  Width of input clock pulse.
- $t_{CKL}$  Interval between clock pulses.
- $t_{CKP1}$  Clock period – when not using Complement array.
- $t_{IS1}$  Required delay between beginning of valid input and positive transition of clock.
- $t_{CKP2}$  Clock period – when using complement array.
- $t_{IS2}$  Required delay between beginning of valid input and positive transition of clock, when using optional Complement Array (two passes necessary through the AND array).
- $t_{VS}$  Required delay between  $V_{CC}$  (after power-on) and negative transition of clock preceding first reliable clock pulse.
- $t_{PRS}$  Required delay between negative transition of asynchronous Preset and negative transition of clock preceding first reliable clock pulse.
- $t_{IH}$  Required delay between positive transition of clock and end of valid input data.
- $t_{CKO}$  Delay between positive transition of clock and when Outputs become valid (with PR/OE Low).
- $t_{OE}$  Delay between beginning of Output Enable Low and when Outputs become valid.
- $t_{OD}$  Delay between beginning of Output Enable High and when Outputs are in the off state.
- $t_{SRE}$  Delay between input  $I_0$  transition to Diagnostic mode and when the Outputs reflect the contents of the State Register.
- $t_{SRD}$  Delay between input  $I_0$  transition to Logic mode and when the Outputs reflect the contents of the Output Register.
- $t_{PR}$  Delay between positive transition of Preset and when Outputs become valid "1".
- $t_{PPR}$  Delay between  $V_{CC}$  (after power-on) and when Outputs become preset at "1".
- $t_{PRH}$  Width of preset input pulse.
- $f_{MAX}$  Maximum clock frequency.

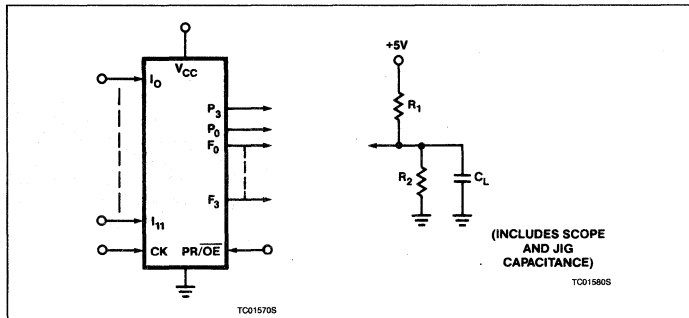
Field-Programmable Logic Sequencer (12 × 48 × 8)

PLS168A

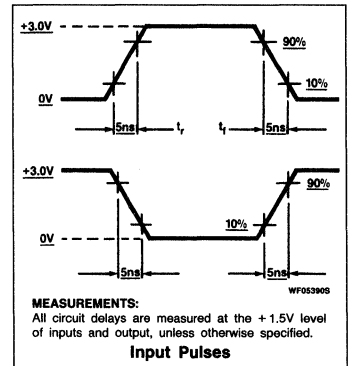
TIMING DIAGRAMS (Continued)



TEST LOAD CIRCUIT



VOLTAGE WAVEFORM



# Field-Programmable Logic Sequencer (12 × 48 × 8)

PLS168A

## SPEED VS. "OR" LOADING

The maximum frequency at which the FPLS can be clocked while operating in *sequential mode* is given by:

$$(1/f_{MAX}) = t_{CY} = t_{IS} + t_{CKO}$$

This frequency depends on the number of transition terms  $T_n$  used. Having all 48 terms connected in the AND array does not appreciably impact performance; but the number of terms connected to each OR line affects  $t_{IS}$ , due to capacitive loading. The effect of this loading can be seen in Figure 1, showing the variation of  $t_{IS1}$  with the number of terms connected per OR.

The AC electrical characteristics contain two limits for the parameters  $t_{IS1}$  and  $t_{IS2}$ . The first,  $t_{IS1A}$  is guaranteed for a device with 24 terms connected to any OR line.  $t_{IS1B}$  is guaranteed for a device with 16 terms connected to any OR line.

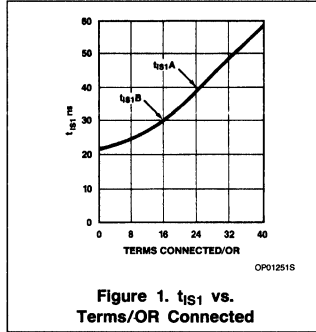


Figure 1.  $t_{IS1}$  vs. Terms/OR Connected

The three other entries in the AC table,  $t_{IS2}$  A and B are corresponding 24 and 16 term limits when using the on-chip Complement Array.

The worst case of  $t_{IS}$  for a given application can be determined by identifying the OR line with the maximum number of  $T_n$  connections. This can be done by referring to the interconnect pattern in the FPLS logic

diagram, typically illustrated in Figure 2, or by counting the maximum number of "H" or "L" entries in one of the columns of the device Program Table.

This number plotted on the curve in Figure 1 will yield the worst case  $t_{IS}$  and, by implication, the maximum clocking frequency for reliable operation.

Note that for maximum speed all UNUSED transition terms should be disconnected from the OR array.

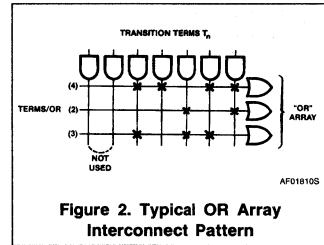


Figure 2. Typical OR Array Interconnect Pattern

# Field-Programmable Logic Sequencer (12 × 48 × 8)

# PLS168A

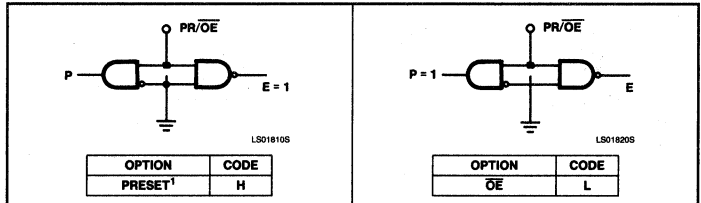
## LOGIC PROGRAMMING

The FPLS can be programmed by means of Logic programming equipment.

With Logic programming, the AND/OR gate input connections necessary to implement the desired logic function are coded directly from the State Diagram using the Program Table on the following page.

In this table, the logic state or action of control variables C, I, P, N, and F, associated with each Transition Term  $T_n$ , is assigned a symbol which results in the proper fusing pattern of corresponding link pairs, defined as follows:

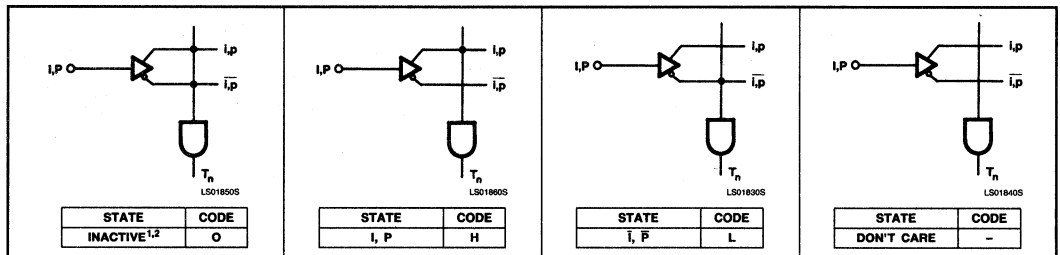
## PRESET/OE OPTION - (P/E)



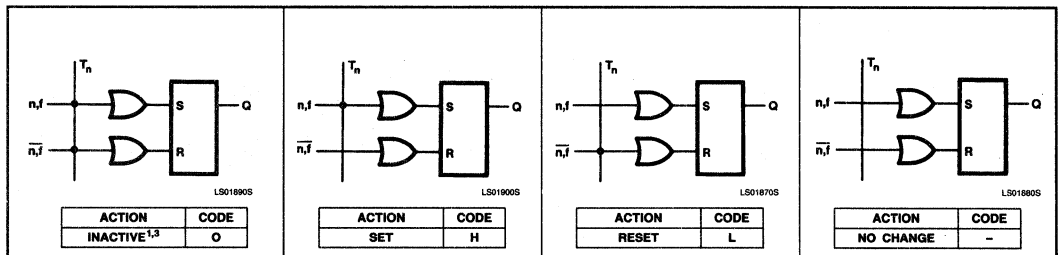
### PROGRAMMING:

The PLS168A has a power-up preset feature. This feature insures that the device will power-up in a known state with all register elements (state and output register) at a logic High (H). When programming the device it is important to realize this is the initial state of the device. You *must* provide a next state jump if you do not wish to use all Highs (H) as the present state.

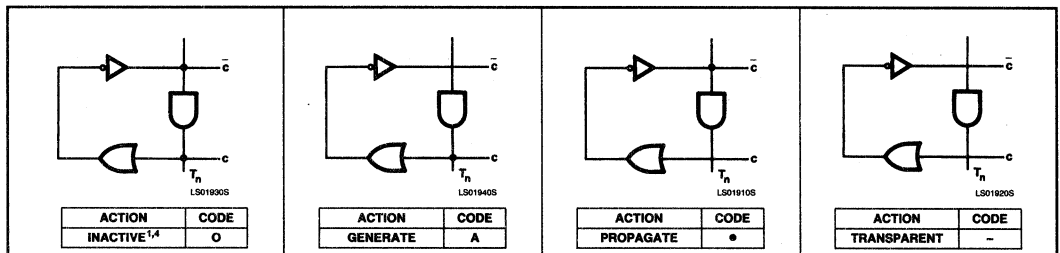
## "AND" ARRAY - (I), (P)



## "OR" ARRAY - (N), (F)



## "COMPLEMENT" ARRAY - (C)



### NOTES:

1. This is the initial unprogrammed state of all link pairs. It is normally associated with all unused (inactive) AND gates  $T_n$ .
2. Any gate  $T_n$  will be unconditionally inhibited if any one of its I or P link pairs is left intact.
3. To prevent simultaneous Set and Reset flip-flop commands, this state is not allowed for N and F link pairs coupled to active gates  $T_n$  (see flip-flop truth tables).
4. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates  $T_n$ .





# PLS173

## Field-Programmable Logic Array (22 × 42 × 10)

Signetics Programmable Logic  
Product Specification

### Application Specific Products

- Series 24

### DESCRIPTION

The PLS173 is a two-level logic element consisting of 42 AND gates and 10 OR gates with fusible link connections for programming I/O polarity and direction.

All AND gates are linked to 12 inputs (I) and 10 bidirectional I/O lines (B). These yield variable I/O gate configurations via 10 direction control gates (D), ranging from 22 inputs to 10 outputs.

On chip T/C buffers couple either True (I, B) or Complement (I, B) input polarities to all AND gates, whose outputs can be optionally linked to all OR gates. Their output polarity, in turn, is individually programmable through a set of EX-OR gates for implementing AND/OR or AND/NOR logic functions.

The PLS173 is field programmable, enabling the user to quickly generate custom patterns using standard programming equipment.

Order codes for this device are contained in the pages following.

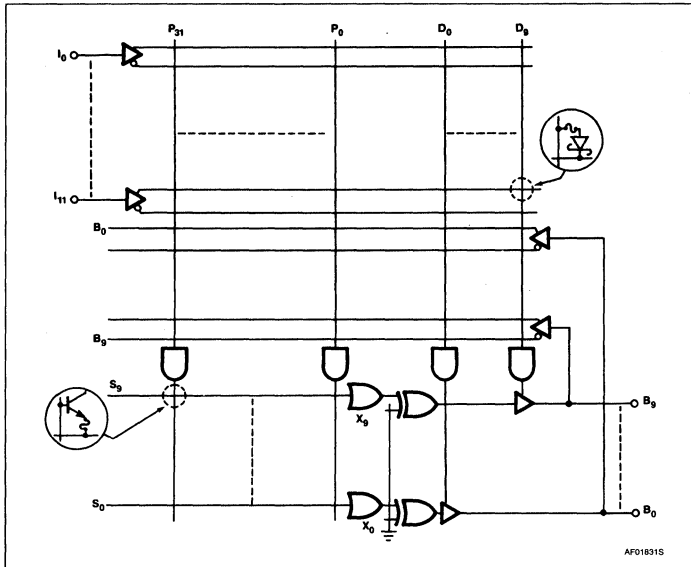
### FEATURES

- Field-Programmable (Ni-Cr links)
- 12 inputs
- 42 AND gates
- 10 OR gates
- 10 bidirectional I/O lines
- Active-High or -Low outputs
- 42 product terms
  - 32 logic terms
  - 10 control terms
- I/O propagation delay: 30ns (max.)
- Input loading: -100μA (max.)
- Power dissipation: 750mW (typ.)
- Tri-state outputs
- TTL compatible

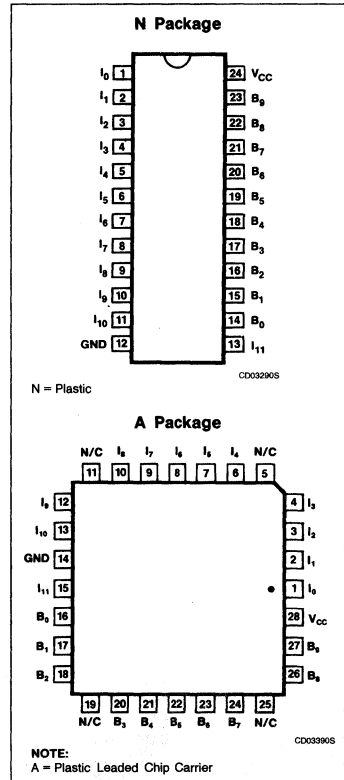
### APPLICATIONS

- Random logic
- Code converters
- Fault detectors
- Function generators
- Address mapping
- Multiplexing

### FUNCTIONAL DIAGRAM



### PIN CONFIGURATIONS



### LOGIC FUNCTION

TYPICAL PRODUCT TERM:

$$P_n = A \cdot \bar{B} \cdot C \cdot D \dots$$

TYPICAL LOGIC FUNCTION:

$$Z = P_0 + P_1 + P_2 \dots$$

AT OUTPUT POLARITY = H

$$Z = P_0 + P_1 + P_2 \dots$$

AT OUTPUT POLARITY = L

$$Z = \bar{P}_0 \cdot \bar{P}_1 \cdot \bar{P}_2 \dots$$

### NOTES:

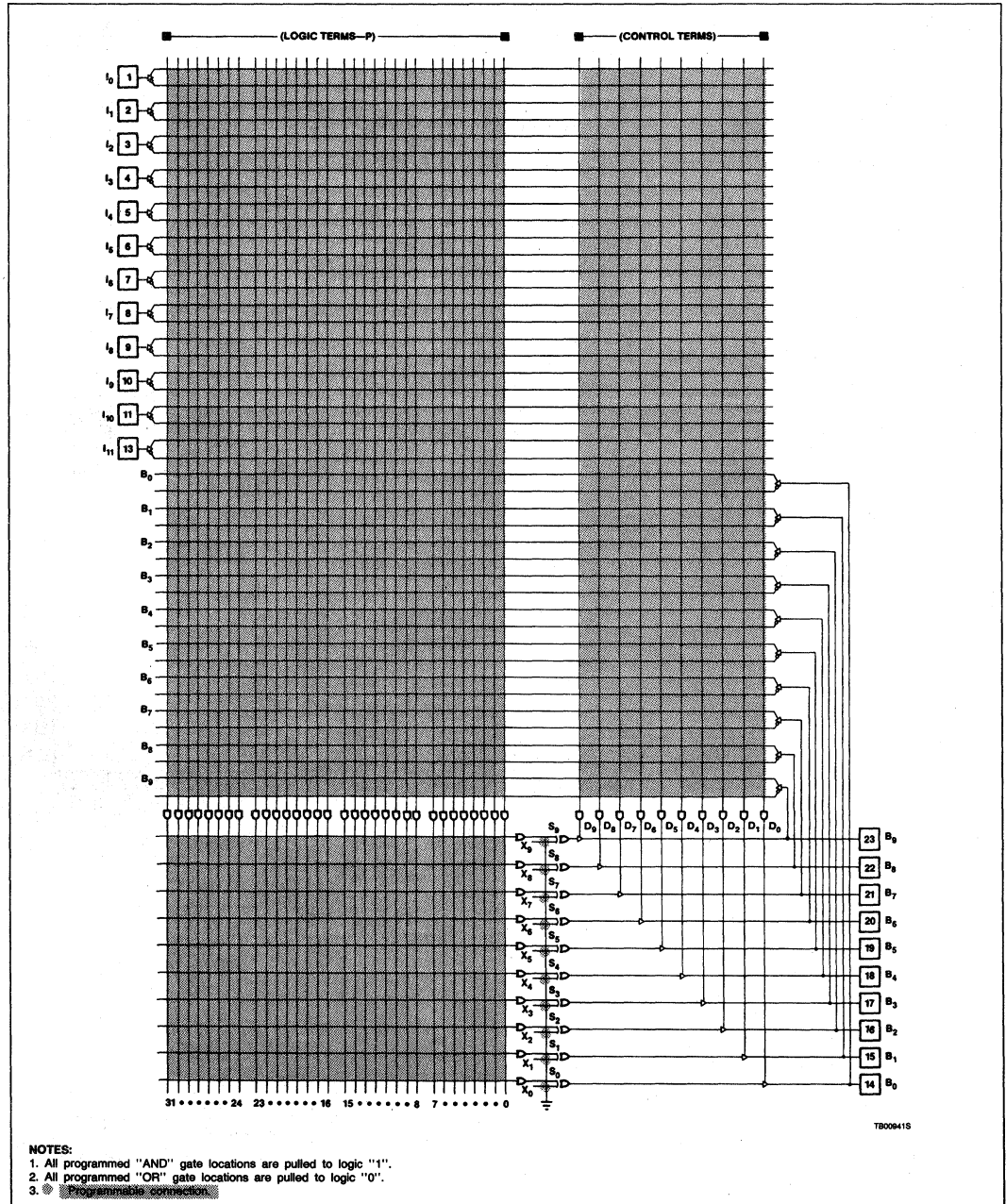
- For each of the 10 outputs, either function Z (Active-High) or Z (Active-Low) is available, but not both. The desired output polarity is programmed via the EX-OR gates.
- Z, A, B, C, etc. are user defined connections to fixed inputs (I), and bidirectional pins (B).



# Field-Programmable Logic Array (22 × 42 × 10)

PLS173

## FPLA LOGIC DIAGRAM



## Field-Programmable Logic Array (22 × 42 × 10)

PLS173

## ORDERING INFORMATION

DESCRIPTION	ORDER CODE
24-pin Plastic DIP 300mil-wide	PLS173N
28-pin Plastic Leaded Chip Carrier	PLS173A

## THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS<sup>1</sup>

SYMBOL	PARAMETER	RATING		UNIT
		Min	Max	
V <sub>CC</sub>	Supply voltage		+7	V <sub>DC</sub>
V <sub>IN</sub>	Input voltage		+5.5	V <sub>DC</sub>
V <sub>OUT</sub>	Output voltage		+5.5	V <sub>DC</sub>
I <sub>IN</sub>	Input currents	-30	+30	mA
I <sub>OUT</sub>	Output currents		+100	mA
T <sub>A</sub> T <sub>STG</sub>	Temperature range Operating Storage	0 -65	+75 +150	°C

DC ELECTRICAL CHARACTERISTICS 0°C ≤ T<sub>A</sub> ≤ +75°C, 4.75 ≤ V<sub>CC</sub> ≤ 5.25V

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ <sup>2</sup>	Max	
<b>Input voltage<sup>3</sup></b>						
V <sub>IL</sub> V <sub>IH</sub> V <sub>IC</sub>	Low High Clamp <sup>3,4</sup>	V <sub>CC</sub> = Min V <sub>CC</sub> = Max V <sub>CC</sub> = Min, I <sub>IN</sub> = -12mA	2.0		0.8 -0.8 -1.2	V
<b>Output voltage</b>						
V <sub>OL</sub> V <sub>OH</sub>	Low <sup>3,5</sup> High <sup>3,6</sup>	V <sub>CC</sub> = Min I <sub>OL</sub> = 15mA I <sub>OH</sub> = -2mA	2.4		0.5	V
<b>Input current<sup>11</sup></b>						
I <sub>IL</sub> I <sub>IH</sub>	Low High	V <sub>CC</sub> = Max V <sub>IN</sub> = 0.45V V <sub>IN</sub> = 5.5V			-100 40	μA
<b>Output current</b>						
I <sub>O(OFF)</sub> I <sub>OS</sub>	Hi-Z state <sup>10</sup> Short circuit <sup>4,6,7</sup>	V <sub>CC</sub> = Max V <sub>OUT</sub> = 5.5V V <sub>OUT</sub> = 0.45V V <sub>OUT</sub> = 0V			80 -140 -70	μA mA
I <sub>CC</sub>	V <sub>CC</sub> supply current <sup>8</sup>	V <sub>CC</sub> = Max		150	170	mA
<b>Capacitance</b>						
I <sub>IN</sub> C <sub>B</sub>	Input I/O	V <sub>CC</sub> = 5V V <sub>IN</sub> = 2.0V V <sub>B</sub> = 2.0V		8 15		pF

5

# Field-Programmable Logic Array (22 × 42 × 10)

PLS173

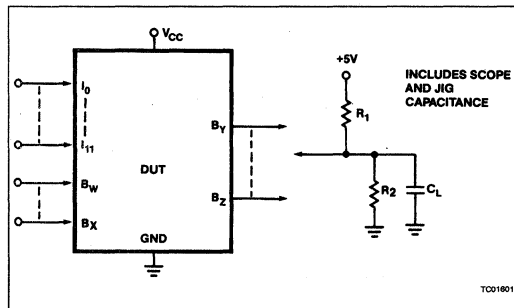
## AC ELECTRICAL CHARACTERISTICS 0°C ≤ T<sub>A</sub> ≤ +75°C, 4.75 ≤ V<sub>CC</sub> ≤ 5.25V, R<sub>1</sub> = 470Ω, R<sub>2</sub> = 1kΩ

SYMBOL	PARAMETER	TO	FROM	TEST CONDITIONS	LIMITS			UNIT
					Min	Typ	Max	
t <sub>PD</sub>	Propagation delay	Output±	Input±	C <sub>L</sub> = 30pF		20	30	ns
t <sub>OE</sub>	Output enable	Output-	Input±	C <sub>L</sub> = 30pf		20	30	ns
t <sub>OD</sub>	Output disable <sup>9</sup>	Output+	Input±	C <sub>L</sub> = 5pF		20	30	ns

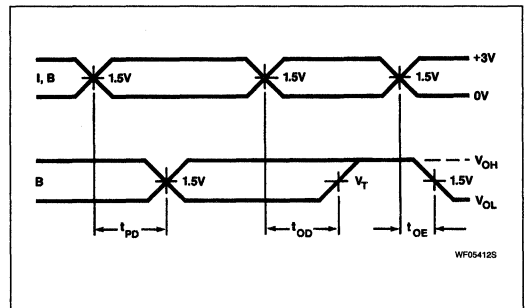
**NOTES:**

- Stresses above those listed under "Absolute Maximum Ratings" may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.
- All typical values are at V<sub>CC</sub> = 5V, T<sub>A</sub> = +25°C.
- All voltage values are with respect to network ground terminal.
- Test one at a time.
- Measured with V<sub>IL</sub> applied to I<sub>11</sub>. Pins 1-5 = 0V, Pins 6-10 = 4.5V, Pin 11 = 0V and Pin 13 = 10V.
- Same conditions as Note 5 except Pin 11 = +10V.
- Duration of short circuit should not exceed 1 second.
- I<sub>CC</sub> is measured with I<sub>0</sub> and I<sub>1</sub> = 0V and I<sub>2</sub>-I<sub>11</sub> and B<sub>0</sub>-B<sub>9</sub> = 4.5V. Part in Virgin State.
- Measured at V<sub>T</sub> = V<sub>OL</sub> + 0.5V.
- Leakage values are a combination of input and output leakage.
- I<sub>IL</sub> and I<sub>IH</sub> limits are for dedicated inputs only (I<sub>0</sub>-I<sub>11</sub>).

### TEST LOAD CIRCUIT



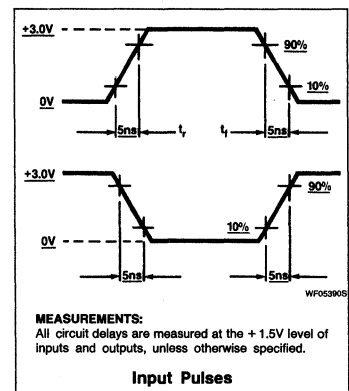
### TIMING DIAGRAM



### TIMING DEFINITIONS

- t<sub>PD</sub> Propagation delay between input and output.
- t<sub>OD</sub> Delay between input change and when output is off (Hi-Z or High).
- t<sub>OE</sub> Delay between input change and when output reflects specified output level.

### VOLTAGE WAVEFORM



# Field-Programmable Logic Array (22 × 42 × 10)

PLS173

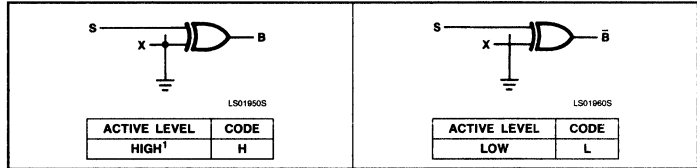
## LOGIC PROGRAMMING

The FPLS can be programmed by means of Logic programming equipment.

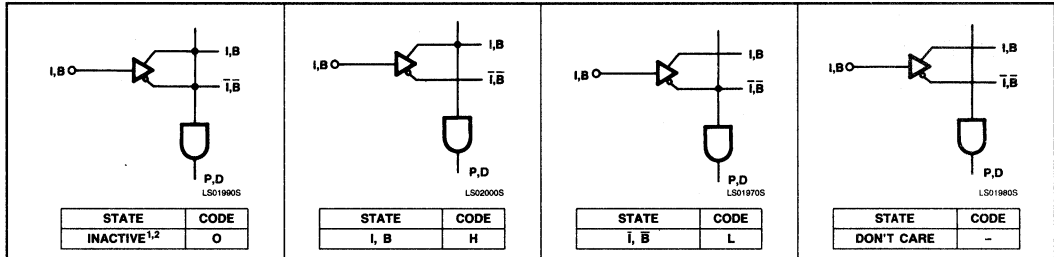
With Logic programming, the AND/OR/EX-OR gate input connections necessary to implement the desired logic function are coded directly from logic equations using the Program Table on the following page.

In this table, the logic state of variables I, P, and B associated with each Sum Term S is assigned a symbol which results in the proper fusing pattern of corresponding link pairs, defined as follows:

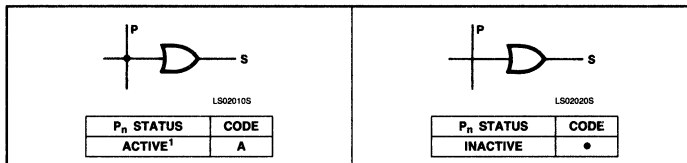
## OUTPUT POLARITY - (B)



## "AND" ARRAY - (I, B)



## OR ARRAY - (B)



### NOTES:

1. This is the initial unprogrammed state of all link pairs. It is normally associated with all unused (inactive) AND gates P<sub>n</sub>, D<sub>n</sub>.
2. Any gate P<sub>n</sub>, D<sub>n</sub> will be unconditionally inhibited if both the true and complement of any input (I, B) are left intact.

## VIRGIN STATE

A factory shipped virgin device contains all fusible links intact, such that:

1. All outputs at "H" polarity.
2. All P<sub>n</sub> terms are disabled.
3. All P<sub>n</sub> terms are active on all outputs.

## CAUTION: PLS173 TEST COLUMNS

The PLS173 incorporates two columns not shown in the logic block diagram. These columns are used for in-house testing of the device in the unprogrammed state. These columns must be disabled prior to using the PLS173 in your application. If you are using a Signetics-approved programmer, the disabling is accomplished during the device programming sequence. If these columns are not disabled, abnormal operation is possible.

5

Field-Programmable Logic Array (22 × 42 × 10)

PLS173

FPLA PROGRAM TABLE

CUSTOMER NAME _____ SIGNETICS DEVICE # _____ PROGRAM TABLE # _____ REV _____ DATE _____		AND																						OR										POLARITY									
		A(1)											B(1)											B(0)																			
		11	10	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	0	0	7	6	5	4	3	2	1	0										
Y																																											
R																																											
B																																											
R																																											
M																																											
E																																											
S																																											
E																																											
R																																											
T																																											
1																																											
2																																											
3																																											
4																																											
5																																											
6																																											
7																																											
8																																											
9																																											
10																																											
11																																											
12																																											
13																																											
14																																											
15																																											
16																																											
17																																											
18																																											
19																																											
20																																											
21																																											
22																																											
23																																											
24																																											
25																																											
26																																											
27																																											
28																																											
29																																											
30																																											
31																																											
D0																																											
D1																																											
D2																																											
D3																																											
D4																																											
D5																																											
D6																																											
D7																																											
D8																																											
D9																																											
D10																																											
D11																																											
D12																																											
D13																																											
D14																																											
D15																																											
D16																																											
D17																																											
D18																																											
D19																																											
D20																																											
D21																																											
D22																																											
D23																																											
D24																																											
D25																																											
D26																																											
D27																																											
D28																																											
D29																																											
D30																																											
D31																																											
D32																																											
D33																																											
D34																																											
D35																																											
D36																																											
D37																																											
D38																																											
D39																																											
D40																																											
D41																																											
D42																																											
D43																																											
D44																																											
D45																																											
D46																																											
D47																																											
D48																																											
D49																																											
D50																																											
D51																																											
D52																																											
D53																																											
D54																																											
D55																																											
D56																																											
D57																																											
D58																																											
D59																																											
D60																																											
D61																																											
D62																																											
D63																																											
D64																																											
D65																																											
D66																																											
D67																																											
D68																																											
D69																																											
D70																																											
D71																																											
D72																																											
D73																																											
D74																																											
D75																																											
D76																																											
D77																																											
D78																																											
D79																																											
D80																																											
D81																																											
D82																																											
D83																																											
D84																																											
D85																																											
D86																																											
D87																																											
D88																																											
D89																																											
D90																																											
D91																																											
D92																																											
D93																																											
D94																																											
D95																																											
D96																																											
D97																																											
D98																																											
D99																																											
D100																																											
D101																																											
D102																																											
D103																																											
D104																																											
D105																																											
D106																																											
D107																																											
D108																																											
D109																																											
D110																																											
D111																																											
D112																																											
D113																																											
D114																																											
D115																																											
D116																																											
D117																																											
D118																																											
D119																																											
D120																																											
D121																																											
D122																																											
D123																																											
D124																																											
D125																																											
D126																																											
D127																																											
D128																																											
D129																																											
D130																																											
D131																																											
D132																																											
D133																																											
D134																																											
D135																																											
D136																																											
D137																																											
D138																																											
D139																																											
D140																																											
D141																																											
D142																																											
D143																																											
D144																																											
D145																																											
D146																																											
D147																																											
D148																																											
D149																																											
D150																																											
D151																																											
D152																																											
D153																																											
D154																																											

# PLS179

## Field-Programmable Logic Sequencer (20 × 45 × 12)

Signetics Programmable Logic  
Product Specification

Application Specific Products  
• Series 24

### DESCRIPTION

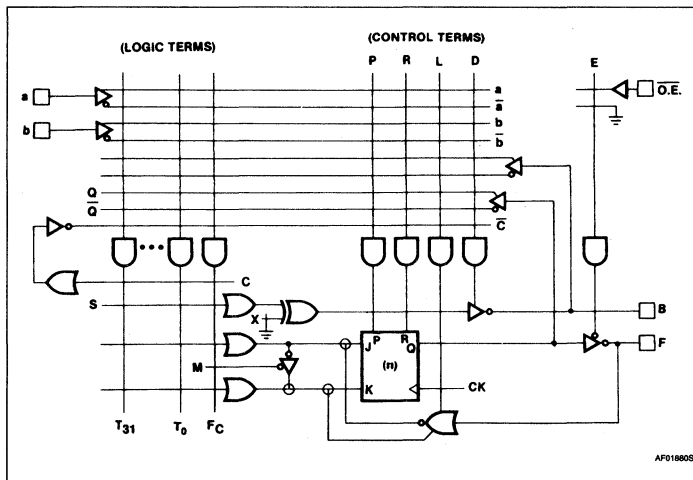
The PLS179 is a Tri-state output, registered logic element combining AND/OR gate arrays with clocked J-K flip-flops. These J-K flip-flops are dynamically convertible to D-type via a "foldback" inverting buffer and control gate  $F_C$ . It features 8 registered I/O outputs (F) in conjunction with 4 bidirectional I/O lines (B). There are 8 dedicated inputs. These yield variable I/O gate and register configurations via control gates (D, L) ranging from 20 inputs to 12 outputs.

The AND/OR arrays consist of 32 logic AND gates, 13 control AND gates, and 21 OR gates with fusible link connections for programming I/O polarity and direction. All AND gates are linked to 8 inputs (I), bidirectional I/O lines (B), internal flip-flop outputs (Q), and Complement Array output (C). The Complement Array consists of a NOR gate optionally linked to all AND gates for generating and propagating complementary AND terms.

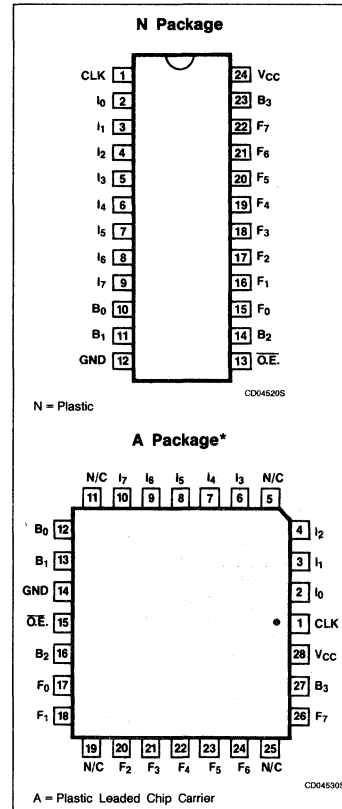
### FEATURES

- Field-Programmable (Ni-Cr link)
- 8 dedicated inputs
- 13 control gates
- 32 AND gates
- 21 OR gates
- 45 product terms:
  - 32 logic terms
  - 13 control terms
- 4 bidirectional I/O lines
- 8 bidirectional registers
- J/K, T, or D-type flip-flops
- Asynchronous Preset/Reset
- Complement Array
- Active-High or -Low outputs
- Programmable OE control
- Positive edge-triggered clock
- Power-on reset on flip-flop ( $F_n = "1"$ )
- Clock frequency: PLS179: 18MHz (max.)
- Input loading: PLS179: - 100 $\mu$ A (max.)
- Power dissipation: 750mW (typ.)
- TTL compatible
- Tri-state option

### FUNCTIONAL DIAGRAM



### PIN CONFIGURATIONS



### APPLICATIONS

- Random sequential logic
- Synchronous up/down counters
- Shift registers
- Bidirectional data buffers
- Timing function generators
- System controllers/synchronizers
- Priority encoder/registers

# Field-Programmable Logic Sequencer (20 × 45 × 12)

PLS179

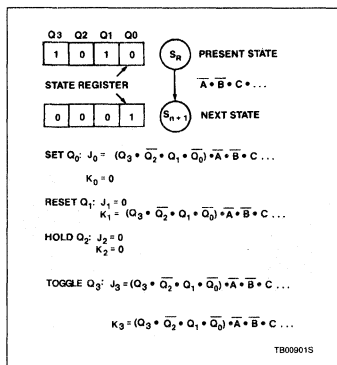
On-chip T/C buffers couple either True (I, B, Q) or Complement (I, B, Q, C) input polarities to all AND gates, whose outputs can be optionally linked to all OR gates. One group of AND gates drives bidirectional I/O lines (B), whose output polarity is individually programmable through a set of EX-OR gates for implementing AND-OR or AND-NOR logic functions. Another group drives the J-K inputs of all flip-flops, as well as asynchronous Preset and Reset lines (P, R).

All flip-flops are positive edge-triggered and can be used as input, output or I/O (for interfacing with a bidirectional data bus) in conjunction with load control gates (L), steering inputs (I), (B), (Q) and programmable output select lines (E).

The PLS179 is field programmable, enabling the user to quickly generate custom patterns using standard programming equipment.

Order codes are contained on the pages following.

## LOGIC FUNCTION



**NOTES:**  
 Similar logic functions are applicable for D and T mode flip-flops.

## FLIP-FLOP TRUTH TABLE

$\bar{O}E$	L	CK	P	R	J	K	Q	F
H								H/Hi-Z
L	X	X	X	X	X	X	L	H
L	X	X	H	L	X	X	H	L
L	X	X	L	H	X	X	L	H
L	L	↑	L	L	L	L	Q	$\bar{Q}$
L	L	↑	L	L	L	H	L	H
L	L	↑	L	L	H	L	H	L
L	L	↑	L	L	H	H	$\bar{Q}$	Q
H	H	↑	L	L	L	H	L	H*
H	H	↑	L	L	H	L	H	L*
+10V	X	↑	X	X	L	H	L	H**
	X	↑	X	X	H	L	H	L**

- NOTES:**
- Positive Logic:  
 $J/K = T_0 + T_1 + T_2 + \dots + T_{31}$   
 $T_n = \bar{C} \cdot (I_0 \cdot I_1 \cdot I_2 \dots) \cdot (Q_0 \cdot Q_1 \dots) \cdot (B_0 \cdot B_1 \dots)$   
 ↑ denotes transition from Low to High level.
  - X = Don't Care
  - \* = Forced at  $F_n$  pin for loading J/K flip-flop in I/O mode. L must be enabled, and other active  $T_n$  disabled via steering input(s) I, B, or Q.
  - At P = R = H, Q = H. The final state of Q depends on which is released first.
  - \*\* = Forced at  $F_n$  pin to load J/K flip-flop independent of program code (Diagnostic mode), Tri-state B outputs.

## VIRGIN STATE

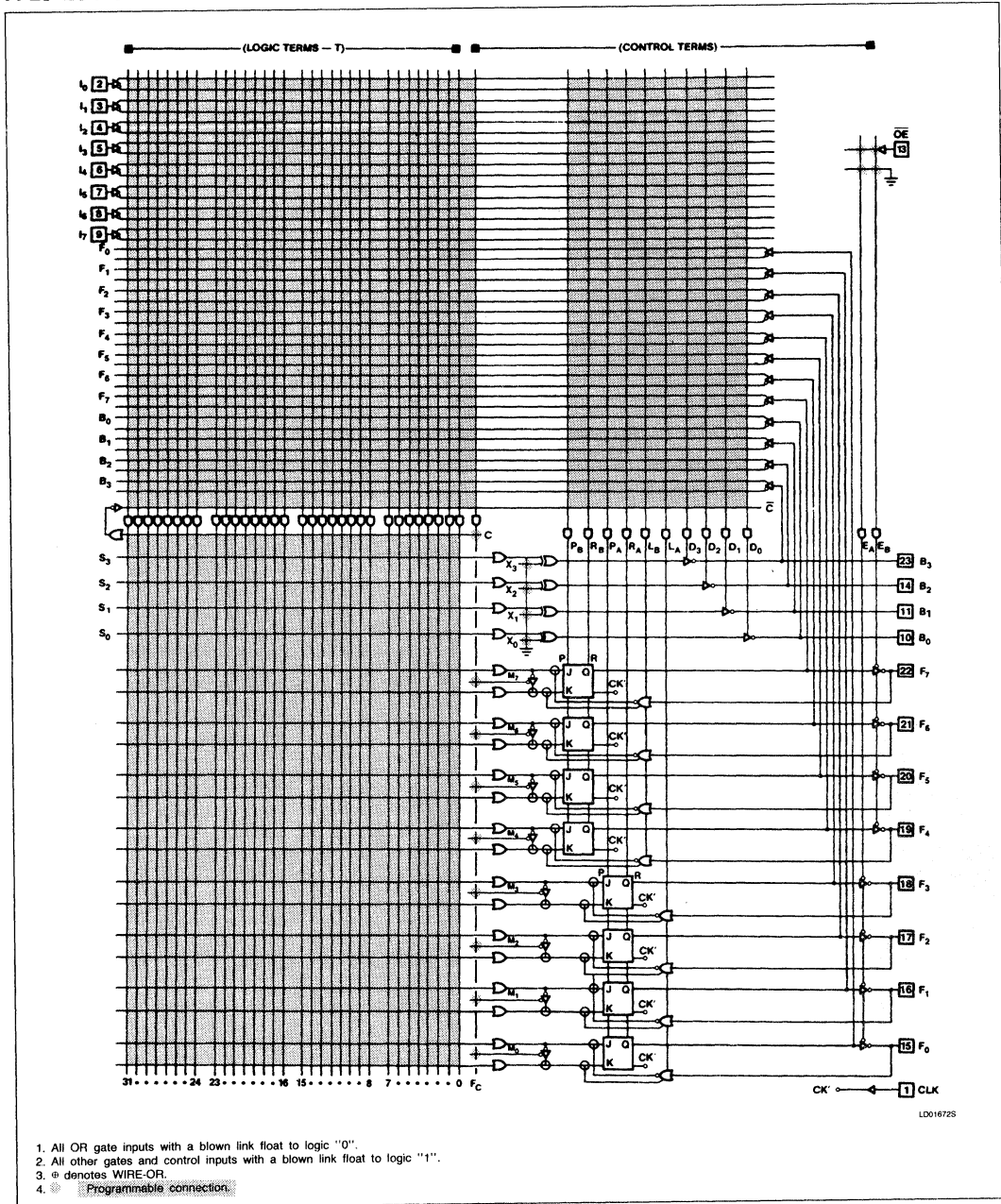
The factory shipped virgin device contains all fusible links intact, such that:

- $\bar{O}E$  is always enabled.
- Preset and Reset are always disabled.
- All transition terms are disabled.
- All flip-flops are in D-mode unless otherwise programmed to J/K only or J/K or D (controlled).
- All B pins are inputs and all F pins are outputs unless otherwise programmed.

# Field-Programmable Logic Sequencer (20 × 45 × 12)

PLS179

## FPLS LOGIC DIAGRAM





## Field-Programmable Logic Sequencer (20 × 45 × 12)

PLS179

## ORDERING INFORMATION

DESCRIPTION	ORDER CODE
24-pin Plastic DIP 300mil-wide	PLS179N
28-pin Plastic Leaded Chip Carrier	PLS179A

## THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS<sup>1</sup>

SYMBOL	PARAMETER	RATING		UNIT
		Min	Max	
V <sub>CC</sub>	Supply voltage		+7	V <sub>DC</sub>
V <sub>IN</sub>	Input voltage		+5.5	V <sub>DC</sub>
V <sub>OUT</sub>	Output voltage		+5.5	V <sub>DC</sub>
I <sub>IN</sub>	Input currents	-30	+30	mA
I <sub>OUT</sub>	Output currents		+100	mA
T <sub>A</sub> T <sub>STG</sub>	Temperature range Operating Storage	0 -65	+75 +150	°C

DC ELECTRICAL CHARACTERISTICS 0°C ≤ T<sub>A</sub> ≤ 75°C, 4.75V ≤ V<sub>CC</sub> ≤ 5.25V

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ <sup>2</sup>	Max	
<b>Input voltage<sup>3</sup></b>						
V <sub>IH</sub> V <sub>IL</sub> V <sub>IC</sub>	High Low Clamp	V <sub>CC</sub> = Max V <sub>CC</sub> = Min V <sub>CC</sub> = Min, I <sub>IN</sub> = -12mA	2		0.8 -1.2	V
<b>Output voltage<sup>3</sup></b>						
V <sub>OH</sub> V <sub>OL</sub>	High Low	V <sub>CC</sub> = Min, I <sub>OH</sub> = -2mA I <sub>OL</sub> = 10mA	2.4	0.35	0.5	V
<b>Input current</b>						
I <sub>IH</sub> I <sub>IL</sub>	High Low	V <sub>CC</sub> = Max, V <sub>IN</sub> = 5.5V V <sub>IN</sub> = 0.45V		< 1 -10	40 -100	μA
<b>Output current</b>						
I <sub>O(OFF)</sub>	Hi-Z state <sup>5,8</sup>	V <sub>CC</sub> = Max, V <sub>OUT</sub> = 5.5V V <sub>OUT</sub> = 0.45V		1	80 -140	μA
I <sub>OS</sub>	Short circuit <sup>4,6</sup>	V <sub>OUT</sub> = 0V	-15		-70	mA
I <sub>CC</sub>	V <sub>CC</sub> supply current <sup>7</sup>	V <sub>CC</sub> = Max		150	210	mA
<b>Capacitance</b>						
C <sub>IN</sub> C <sub>OUT</sub>	Input Output	V <sub>CC</sub> = 5.0V, V <sub>IN</sub> = 2.0V V <sub>OUT</sub> = 2.0V		8 15		pF

## NOTES:

- Stresses above those listed under "Absolute Maximum Ratings" may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.
- All typical values are at V<sub>CC</sub> = 5V, T<sub>A</sub> = +25°C.
- All voltage values are with respect to network ground terminal.
- Test one at a time.
- Measured with V<sub>IH</sub> applied to  $\overline{OE}$ .
- Duration of short circuit should not exceed 1 second.
- I<sub>CC</sub> is measured with the  $\overline{OE}$  input grounded, all other inputs at 4.5V, and the outputs open.
- Leakage values are a combination of input and output leakage.

# Field-Programmable Logic Sequencer (20 × 45 × 12)

PLS179

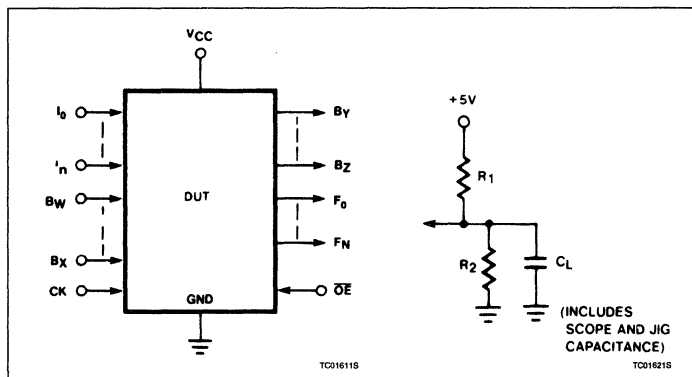
## AC ELECTRICAL CHARACTERISTICS $R_1 = 470\Omega$ , $R_2 = 1k\Omega$ , $0^\circ C \leq T_A \leq +75^\circ C$ , $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TO	FROM	TEST CONDITIONS	LIMITS			UNIT
					Min <sup>5</sup>	Typ <sup>1</sup>	Max	
<b>Pulse width</b>								
$t_{CKH}$	Clock <sup>2</sup> high	CK -	CK +	$C_L = 30pF$	20	15		ns
$t_{CKL}$	Clock low	CK +	CK -		20	15		
$t_{CKP}$	Period	CK +	CK +		55	45		
$t_{PRH}$	Preset/Reset pulse	(I,B) +	(I,B) -		35	30		
<b>Setup time</b>								
$t_{IS1}$	Input	CK +	(I,B) $\pm$	$C_L = 30pF$	35	30		ns
$t_{IS2}$	Input (through $F_n$ )	CK +	F $\pm$		15	10		
$t_{IS3}$	Input (through Complement Array) <sup>4</sup>	CK +	(I,B) $\pm$		55	45		
<b>Hold time</b>								
$t_{IH1}$	Input	CK +	(I,B) $\pm$	$C_L = 30pF$	0	-5		ns
$t_{IH2}$	Input (through $F_n$ )	CK +	F $\pm$		15	10		
<b>Propagation delay</b>								
$t_{CKO}$	Clock	F $\pm$	CK $\pm$	$C_L = 5pF$		15	20	ns
$t_{OE1}$	Output enable	F -	$\overline{OE}$ -			20	30	
$t_{OD1}$	Output disable <sup>3</sup>	F +	$\overline{OE}$ +	$C_L = 30pF$		20	30	
$t_{PD}$	Output	B $\pm$	(I,B) $\pm$			25	35	
$t_{OE2}$	Output enable	B $\pm$	(I,B) +	$C_L = 5pF$		20	30	
$t_{OD2}$	Output disable <sup>3</sup>	B +	(I,B) -			20	30	
$t_{PRO}$	Preset/Reset	F $\pm$	(I,B) +	$C_L = 30pF$		35	45	
$t_{PPR}$	Power-on preset	F -	$V_{CC}$ +			0	10	

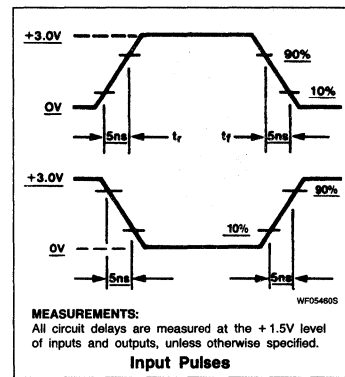
**NOTES:**

1. All typical values are at  $V_{CC} = 5V$ ,  $T_A = +25^\circ C$ .
2. To prevent spurious clocking, clock rise time (10% - 90%)  $\leq 10ns$ .
3. Measured at  $V_T = V_{OL} + 0.5V$ .
4. When using the Complement Array  $T_{CKP} = 75ns$  (min).
5. Limits are guaranteed with 12 product terms maximum connected to each sum term line.

**TEST LOAD CIRCUIT**



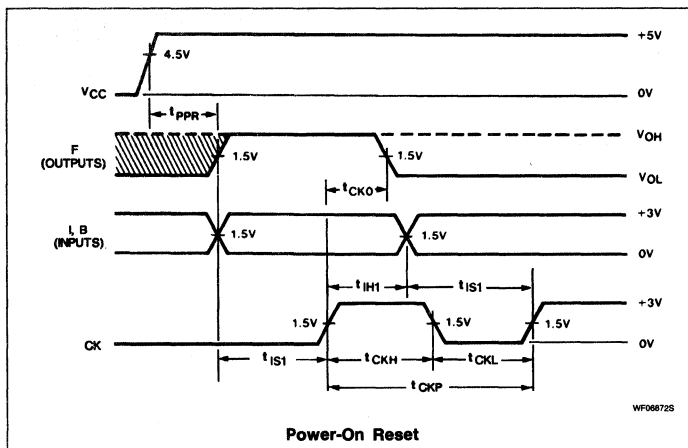
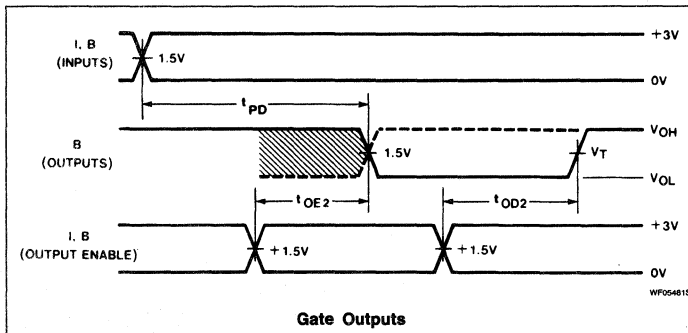
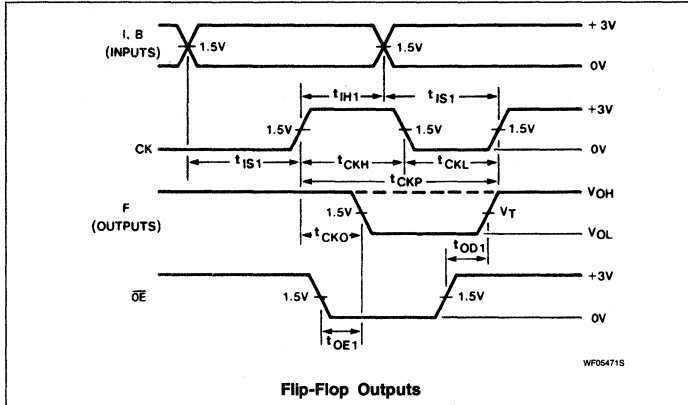
**VOLTAGE WAVEFORM**



# Field-Programmable Logic Sequencer (20 × 45 × 12)

PLS179

## TIMING DIAGRAMS



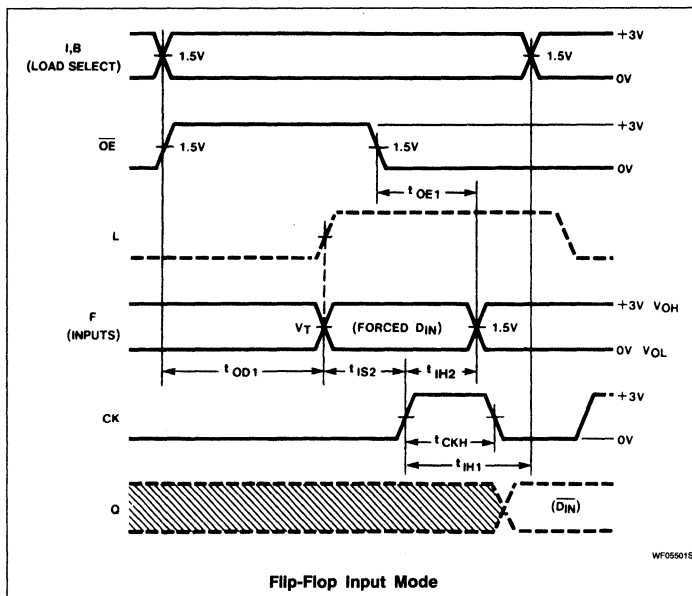
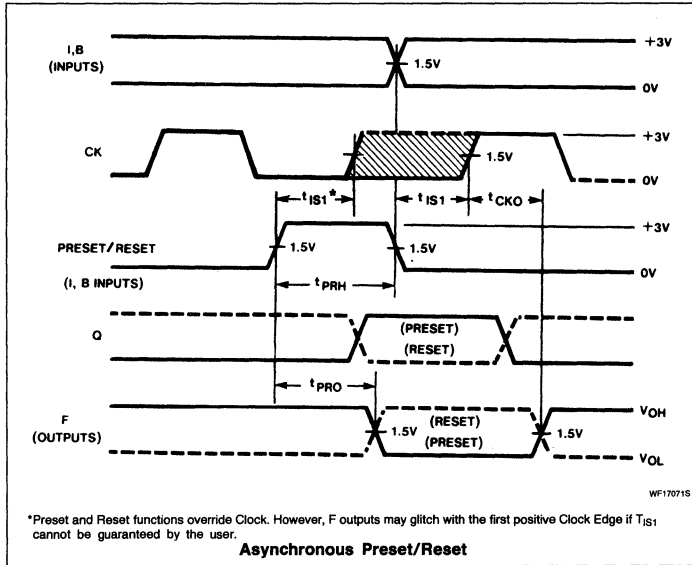
## TIMING DEFINITIONS

- $t_{CKH}$  Width of input clock pulse.
- $t_{CKL}$  Interval between clock pulses.
- $t_{CKP}$  Clock period.
- $t_{PRH}$  Width of preset input pulse.
- $t_{IS1}$  Required delay between beginning of valid input and positive transition of clock.
- $t_{IS2}$  Required delay between beginning of valid input forced at flip-flop output pins, and positive transition of clock.
- $t_{IH1}$  Required delay between positive transition of clock and end of valid input data.
- $t_{IH2}$  Required delay between positive transition of clock and end of valid input data forced at flip-flop output pins.
- $t_{CKO}$  Delay between positive transition of clock and when Outputs become valid (with  $\overline{OE}$  Low).
- $t_{OE1}$  Delay between beginning of Output Enable Low and when Outputs become valid.
- $t_{OD1}$  Delay between beginning of Output Enable High and when Outputs are in the off state.
- $t_{PPR}$  Delay between  $V_{CC}$  (after power-on) and when flip-flop outputs become preset at "1" (internal Q outputs at "0").
- $t_{PD}$  Propagation delay between combinational inputs and outputs.
- $t_{OE2}$  Delay between predefined Output Enable High, and when combinational Outputs become valid.
- $t_{OD2}$  Delay between predefined Output Enable Low and when combinational Outputs are in the off state.
- $t_{PRO}$  Delay between positive transition of predefined Preset/Reset input, and when flip-flop outputs become valid.

# Field-Programmable Logic Sequencer (20 × 45 × 12)

PLS179

## TIMING DIAGRAMS (Continued)



# Field-Programmable Logic Sequencer (20 × 45 × 12)

PLS179

The FPLS can be programmed by means of Logic Programming equipment.

With Logic programming, the AND/OR-EX-OR input connections necessary to imple-

ment the desired logic function are coded directly from the State Diagram using the Program Tables on the following pages.

In these Tables, the logic state or action of all I/O, control, and state variables is assigned a symbol which results in the proper fusing pattern of corresponding links defined as follows:

## "AND" ARRAY - (I), (B), (Qp)

<p style="text-align: center;">(T, F<sub>C</sub>, L, P, R, D)<sub>n</sub></p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr><th>STATE</th><th>CODE</th></tr> </thead> <tbody> <tr><td>INACTIVE<sup>1,2</sup></td><td>0</td></tr> </tbody> </table> <p style="text-align: center; font-size: small;">LS920905</p>	STATE	CODE	INACTIVE <sup>1,2</sup>	0	<p style="text-align: center;">(T, F<sub>C</sub>, L, P, R, D)<sub>n</sub></p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr><th>STATE</th><th>CODE</th></tr> </thead> <tbody> <tr><td>I, B, Q</td><td>H</td></tr> </tbody> </table> <p style="text-align: center; font-size: small;">LS920905</p>	STATE	CODE	I, B, Q	H	<p style="text-align: center;">(T, F<sub>C</sub>, L, P, R, D)<sub>n</sub></p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr><th>STATE</th><th>CODE</th></tr> </thead> <tbody> <tr><td>I-bar, B-bar</td><td>L</td></tr> </tbody> </table> <p style="text-align: center; font-size: small;">LS920905</p>	STATE	CODE	I-bar, B-bar	L	<p style="text-align: center;">(T, F<sub>C</sub>, L, P, R, D)<sub>n</sub></p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr><th>STATE</th><th>CODE</th></tr> </thead> <tbody> <tr><td>DON'T CARE</td><td>-</td></tr> </tbody> </table> <p style="text-align: center; font-size: small;">LS920905</p>	STATE	CODE	DON'T CARE	-
STATE	CODE																		
INACTIVE <sup>1,2</sup>	0																		
STATE	CODE																		
I, B, Q	H																		
STATE	CODE																		
I-bar, B-bar	L																		
STATE	CODE																		
DON'T CARE	-																		

## "COMPLEMENT" ARRAY - (C)

<p style="text-align: center;">(T<sub>n</sub>, F<sub>C</sub>)</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr><th>ACTION</th><th>CODE</th></tr> </thead> <tbody> <tr><td>INACTIVE<sup>1,3,5</sup></td><td>0</td></tr> </tbody> </table> <p style="text-align: center; font-size: small;">LS920705</p>	ACTION	CODE	INACTIVE <sup>1,3,5</sup>	0	<p style="text-align: center;">(T<sub>n</sub>, F<sub>C</sub>)</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr><th>ACTION</th><th>CODE</th></tr> </thead> <tbody> <tr><td>GENERATE<sup>5</sup></td><td>A</td></tr> </tbody> </table> <p style="text-align: center; font-size: small;">LS920905</p>	ACTION	CODE	GENERATE <sup>5</sup>	A	<p style="text-align: center;">(T<sub>n</sub>, F<sub>C</sub>)</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr><th>ACTION</th><th>CODE</th></tr> </thead> <tbody> <tr><td>PROPAGATE</td><td>•</td></tr> </tbody> </table> <p style="text-align: center; font-size: small;">LS920905</p>	ACTION	CODE	PROPAGATE	•	<p style="text-align: center;">(T<sub>n</sub>, F<sub>C</sub>)</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr><th>ACTION</th><th>CODE</th></tr> </thead> <tbody> <tr><td>TRANSPARENT</td><td>-</td></tr> </tbody> </table> <p style="text-align: center; font-size: small;">LS921005</p>	ACTION	CODE	TRANSPARENT	-
ACTION	CODE																		
INACTIVE <sup>1,3,5</sup>	0																		
ACTION	CODE																		
GENERATE <sup>5</sup>	A																		
ACTION	CODE																		
PROPAGATE	•																		
ACTION	CODE																		
TRANSPARENT	-																		

## "OR" ARRAY - (MODE)

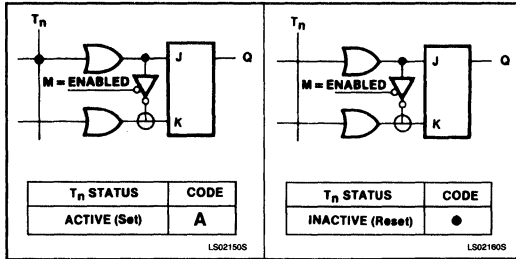
<p style="text-align: center;">(D<sub>n</sub>, L<sub>n</sub>, P<sub>n</sub>, R<sub>n</sub>)</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr><th>ACTION</th><th>CODE</th></tr> </thead> <tbody> <tr><td>PROPAGATE</td><td>•</td></tr> </tbody> </table> <p style="text-align: center; font-size: small;">LS921105</p>	ACTION	CODE	PROPAGATE	•	<p style="text-align: center;">(D<sub>n</sub>, L<sub>n</sub>, P<sub>n</sub>, R<sub>n</sub>)</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr><th>ACTION</th><th>CODE</th></tr> </thead> <tbody> <tr><td>TRANSPARENT</td><td>-</td></tr> </tbody> </table> <p style="text-align: center; font-size: small;">LS921205</p>	ACTION	CODE	TRANSPARENT	-	<p style="text-align: center;">(D<sub>n</sub>, L<sub>n</sub>, P<sub>n</sub>, R<sub>n</sub>)</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr><th>ACTION</th><th>CODE</th></tr> </thead> <tbody> <tr><td>J/K OR D (CONTROLLED)</td><td>A</td></tr> </tbody> </table> <p style="text-align: center; font-size: small;">LS921305</p>	ACTION	CODE	J/K OR D (CONTROLLED)	A	<p style="text-align: center;">(D<sub>n</sub>, L<sub>n</sub>, P<sub>n</sub>, R<sub>n</sub>)</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr><th>ACTION</th><th>CODE</th></tr> </thead> <tbody> <tr><td>J-K</td><td>•</td></tr> </tbody> </table> <p style="text-align: center; font-size: small;">LS921405</p>	ACTION	CODE	J-K	•
ACTION	CODE																		
PROPAGATE	•																		
ACTION	CODE																		
TRANSPARENT	-																		
ACTION	CODE																		
J/K OR D (CONTROLLED)	A																		
ACTION	CODE																		
J-K	•																		

Notes on following page.

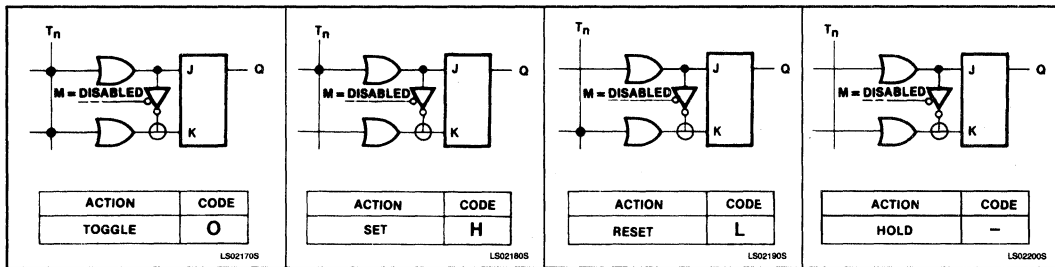
# Field-Programmable Logic Sequencer (20 × 45 × 12)

PLS179

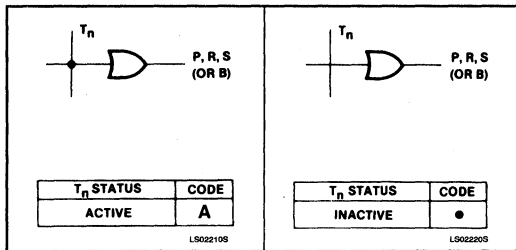
## "OR" ARRAY — ( $Q_N = D$ - Type)



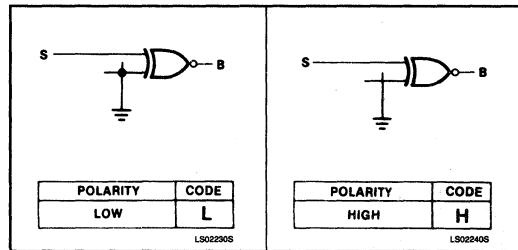
## "AND" ARRAY — ( $Q_N = J - K$ Type)



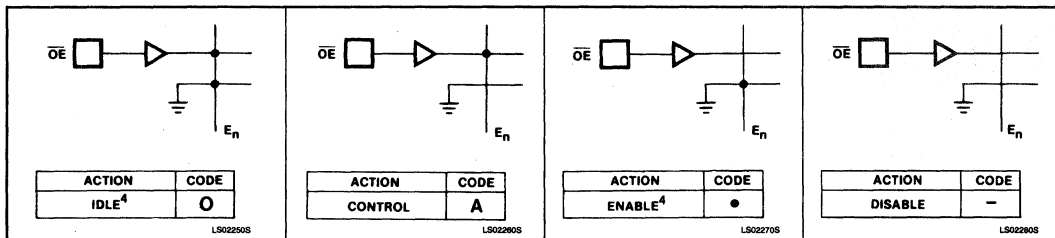
## "OR" ARRAY — (S or B), (P), (R)



## "EX-OR" ARRAY — (B)



## "OE" ARRAY — (E)



**NOTES:**

1. This is the initial unprogrammed state of all link pairs. It is normally associated with all unused (inactive) AND gates.
2. Any gate ( $T_n, F_C, L, P, R, D_n$ ) will be unconditionally inhibited if any one of the I, B, or Q link pairs is left intact.
3. To prevent oscillations, this state is not allowed for C link, pairs coupled to active gates  $T_n, F_C$ .
4.  $E_n = O$  and  $E_n = \bullet$  are logically equivalent states, since both cause  $F_n$  outputs to be unconditionally enabled.
5. These states are not allowed for control gates ( $L, P, R, D_n$ ) due to their lack of "OR" array links.

5

# Field-Programmable Logic Sequencer (20 × 45 × 12)

PLS179

## FPLS PROGRAM TABLE

AND				OR				CONTROL				NOTES
INACTIVE	0	I, B, Q	H	ACTIVE	A	R, R, B (O)	(O = D)	J/K	A	F/F MODE	IDLE	
I, B, Q	H	I, B, Q	H	INACTIVE	•	(Q = D)		J/K or D (CONTROLLED)	A	MODE	CONTROL	A
DON'T CARE	-	Q, B (I)	P	TOGGLE	0	(Q = J/K)		HIGH	H	(POL.)	ENABLE	•
INACTIVE	0	C		SET	H			LOW	L		DISABLE	-
GENERATE	A			RESET	L						E <sub>A</sub> , B	
PROPAGATE	•			HOLD	-							
TRANSPARENT	-											

T E R M C	AND																F/F MODE		E <sub>B</sub>		E <sub>A</sub>		POLARITY	
	B (I)								Q (P)								(OR)							
	7 6 5 4 3 2 1 0				3 2 1 0				7 6 5 4 3 2 1 0				7 6 5 4 3 2 1 0				7 6 5 4 3 2 1 0							
	Q (N)				B (O)																			
0																								
1																								
2																								
3																								
4																								
5																								
6																								
7																								
8																								
9																								
10																								
11																								
12																								
13																								
14																								
15																								
16																								
17																								
18																								
19																								
20																								
21																								
22																								
23																								
24																								
25																								
26																								
27																								
28																								
29																								
30																								
31																								
F <sub>C</sub>																								
P <sub>B</sub>																								
R <sub>B</sub>																								
L <sub>B</sub>																								
P <sub>A</sub>																								
R <sub>A</sub>																								
L <sub>A</sub>																								
D <sub>3</sub>																								
D <sub>2</sub>																								
D <sub>1</sub>																								
D <sub>0</sub>																								
PIN	9	8	7	6	5	4	3	2	23	14	11	10	22	21	20	19	18	17	16	15				

THIS PORTION TO BE COMPLETED BY SIGNETICS

CF (XXXX) \_\_\_\_\_

CUSTOMER SYMBOLIZED PART # \_\_\_\_\_

DATE RECEIVED \_\_\_\_\_

COMMENTS \_\_\_\_\_

CUSTOMER NAME \_\_\_\_\_

PURCHASE ORDER # \_\_\_\_\_

SIGNETICS DEVICE # \_\_\_\_\_

TOTAL NUMBER OF PARTS \_\_\_\_\_

PROGRAM TABLE # \_\_\_\_\_ REV \_\_\_\_\_ DATE \_\_\_\_\_

TB017115

## PLHS473 Field-Programmable Logic Array (20 × 24 × 11)

Signetics Programmable Logic  
Preliminary Specification

### Application Specific Products • Series 24

#### DESCRIPTION

The PLHS473 is a two level logic device consisting of 24 AND gates and 22 OR gates with fusible link connections for programming I/O polarity and direction. The Signetics state of the art Oxide-Isolated Bipolar process is used to produce performance not yet achieved in devices of this complexity.

All AND gates are linked to 11 input pins, 9 bidirectional I/O pins, and 2 dedicated output pins. The bidirectional pins are controlled via the OR array. Using these features, the PLHS473 can be configured with up to 20 inputs and as many as 11 outputs.

The AND array input buffers provide both the True and Complement of the inputs ( $I_x$ ) and the bidirectional signals ( $B_x$ ) as programmable connections to the AND gates. All 24 AND gates can then be optionally linked to all 22 OR gates (a feature known as Product Term sharing not found in PALs<sup>1</sup> or most macrocell architectures). The OR array drives 11 output buffers which can be programmed as Active-High for AND-OR functions or Active-Low for AND-NOR functions. In addition, the I/O configuration of each bidirectional pin is individually controlled by a sum-of-products (AND-OR) function which may also con-

tain any of the 24 AND gate outputs. This allows dynamic I/O configuration of all 9 bidirectional pins.

The PLHS473 contains two new features of significance. A code verification lock has been incorporated to improve user security. The addition of three test columns and one test row enables the user to test the device in an unprogrammed state.

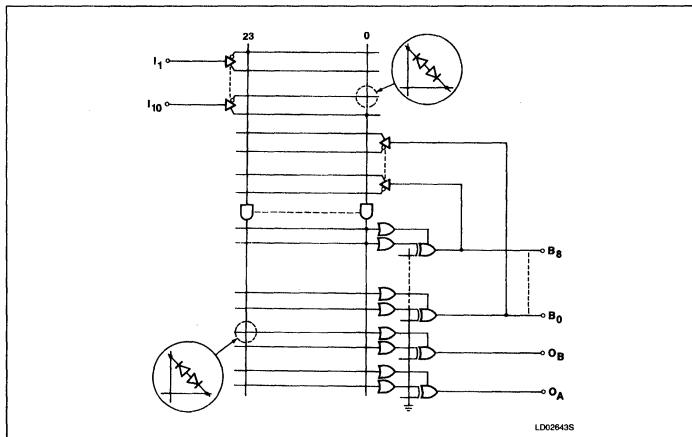
The PLHS473 is field programmable using Vertical Avalanche Migration Programmed (VAMP™) fuses to program the cells. This enables the generation of custom logic patterns using standard programming equipment.

Order codes for this device are contained in the pages following.

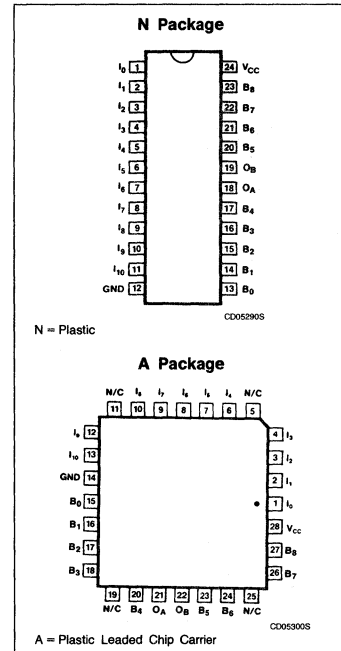
#### FEATURES

- Field-Programmable
- 11 dedicated inputs
- 2 dedicated outputs
- 9 bidirectional I/O lines
- 24 product terms
- 22 OR gates
- I/O direction decoded in OR array
- Output Enable decoded in OR array

#### FUNCTIONAL DIAGRAM



#### PIN CONFIGURATIONS



- I/O propagation delay: 20ns (max.)
- Input loading: -100μA (max.)
- Power dissipation: 700mW (typ.)
- Security fuse
- Testable in unprogrammed state
- Programmable as Tri-state or Open-Collector outputs
- TTL compatible

#### APPLICATIONS

- Random logic
- Code converters
- Fault detectors
- Function generators
- Address mapping
- Multiplexing

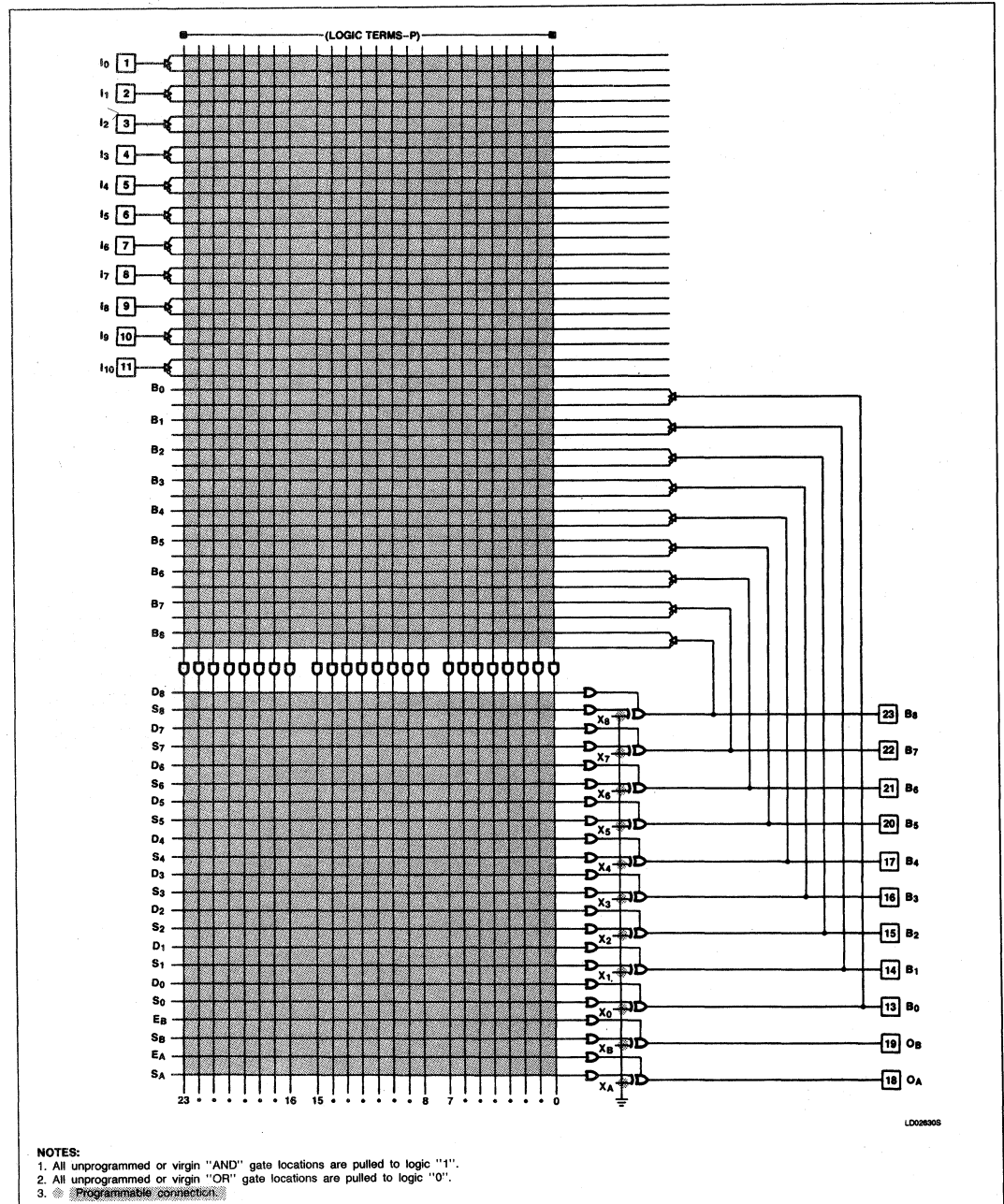
<sup>1</sup> PAL is a trademark of MMI.



# Field-Programmable Logic Array (20 × 24 × 11)

PLHS473

## FPLA LOGIC DIAGRAM



# Field-Programmable Logic Array (20 × 24 × 11)

PLHS473

## ORDERING INFORMATION

DESCRIPTION	ORDER CODE
24-pin Plastic DIP 300mil-wide	PLHS473N
28-pin Plastic Leaded Chip Carrier	PLHS473A

## ABSOLUTE MAXIMUM RATINGS<sup>1</sup>

SYMBOL	PARAMETER	RATING		UNIT
		Min	Max	
V <sub>CC</sub>	Supply voltage		+7	V <sub>DC</sub>
V <sub>IN</sub>	Input voltage		+5.5	V <sub>DC</sub>
V <sub>OUT</sub>	Output voltage		+5.5	V <sub>DC</sub>
I <sub>IN</sub>	Input currents	-30	+30	mA
I <sub>OUT</sub>	Output currents		+100	mA
T <sub>A</sub> T <sub>STG</sub>	Temperature range Operating Storage	0 -65	+75 +150	°C

## LOGIC FUNCTION

**TYPICAL PRODUCT TERM:**  
Pn = A · B · C · D · . . .

**TYPICAL LOGIC FUNCTION:**  
AT OUTPUT POLARITY = H  
Z = P0 + P1 + P2 . . .

AT OUTPUT POLARITY = L  
Z = P0 + P1 + P2 + . . .  
Z = P0 · P1 · P2 · . . .

**NOTES:**  
1. For each of the 11 outputs, either function Z (Active-High) or Z̄ (Active-Low) is available, but not both. The desired output polarity is programmed via the EX-OR gates.  
2. Z, A, B, C, etc. are user defined connections to fixed inputs (I), fixed output pins (O) and bidirectional pins (B).

## THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

## DC ELECTRICAL CHARACTERISTICS 0°C ≤ T<sub>A</sub> ≤ +75°C, 4.75 ≤ V<sub>CC</sub> ≤ 5.25V

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ <sup>2</sup>	Max	
<b>Input voltage<sup>3</sup></b>						
V <sub>IL</sub> V <sub>IH</sub> V <sub>IC</sub>	Low High Clamp <sup>4</sup>	V <sub>CC</sub> = Min V <sub>CC</sub> = Max V <sub>CC</sub> = Min, I <sub>IN</sub> = -12mA	2.0	-0.8	0.8 -1.2	V
<b>Output voltage<sup>3</sup></b>						
V <sub>OL</sub> V <sub>OH</sub>	Low <sup>5</sup> High <sup>6</sup>	V <sub>CC</sub> = Min I <sub>OL</sub> = 15mA I <sub>OH</sub> = -2mA	2.4		0.5	V
<b>Input current</b>						
I <sub>IL</sub> I <sub>IH</sub>	Low High	V <sub>CC</sub> = Max V <sub>IN</sub> = 0.45V V <sub>IN</sub> = 5.5V			-100 40	μA
<b>Output current</b>						
I <sub>O(OFF)</sub> I <sub>OS</sub>	Hi-Z state <sup>10</sup> Short circuit <sup>4,6,7</sup>	V <sub>CC</sub> = Max V <sub>OUT</sub> = 5.5V V <sub>OUT</sub> = 0V	-15		40 -100 -70	μA mA
I <sub>CC</sub>	V <sub>CC</sub> supply current <sup>8</sup>	V <sub>CC</sub> = Max		140	155	mA
<b>Capacitance</b>						
I <sub>IN</sub> C <sub>B</sub>	Input I/O	V <sub>CC</sub> = 5V V <sub>IN</sub> = 2.0V V <sub>B</sub> = 2.0V		8 15		pF

**5**

# Field-Programmable Logic Array (20 × 24 × 11)

PLHS473

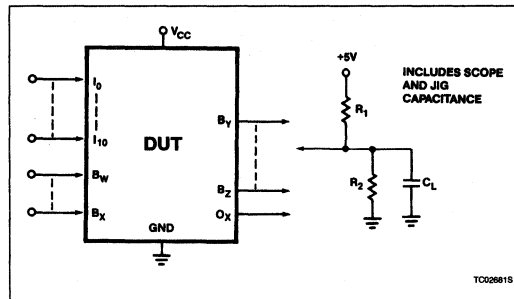
## AC ELECTRICAL CHARACTERISTICS $0^{\circ}\text{C} \leq T_A \leq +75^{\circ}\text{C}$ , $4.75 \leq V_{CC} \leq 5.25\text{V}$ , $R_1 = 470\Omega$ , $R_2 = 1\text{k}\Omega$

SYMBOL	PARAMETER	TO	FROM	TEST CONDITIONS	LIMITS			UNIT
					Min	Typ	Max	
$t_{PD}$	Propagation delay	Output±	Input±	$C_L = 30\text{pF}$		15	20	ns
$t_{OE}$	Output enable	Output-	Input±	$C_L = 30\text{pF}$		15	20	ns
$t_{OD}$	Output disable <sup>9</sup>	Output+	Input±	$C_L = 5\text{pF}$		15	20	ns

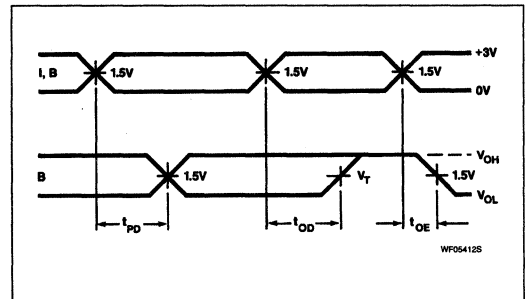
**NOTES:**

- Stresses above those listed under "Absolute Maximum Ratings" may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.
- All typical values are at  $V_{CC} = 5\text{V}$ ,  $T_A = +25^{\circ}\text{C}$ .
- All voltage values are with respect to network ground terminal.
- Test one at a time.
- Measured with Pins 1 - 5 = 0V, Pins 6, 8 = 4.5V, and Pins 7, 9 - 11 = 10V.
- Same conditions as Note 5 except Pin 9 = 4.5V.
- Duration of short circuit should not exceed 1 second.
- $I_{CC}$  is measured with all inputs and bidirectional pins at 4.5V. Part in Virgin State.
- Measured at  $V_T = V_{OL} + 0.5\text{V}$ .
- Leakage values are a combination of input and output leakage.

**TEST LOAD CIRCUIT**



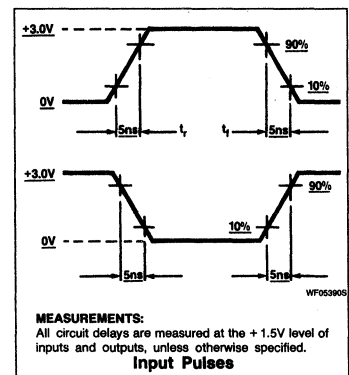
**TIMING DIAGRAM**



**TIMING DEFINITIONS**

- $t_{PD}$  Propagation delay between input and output.
- $t_{OD}$  Delay between input change and when output is off (Hi-Z or High).
- $t_{OE}$  Delay between input change and when output reflects specified output level.

**VOLTAGE WAVEFORM**



# Field-Programmable Logic Array (20 × 24 × 11)

PLHS473

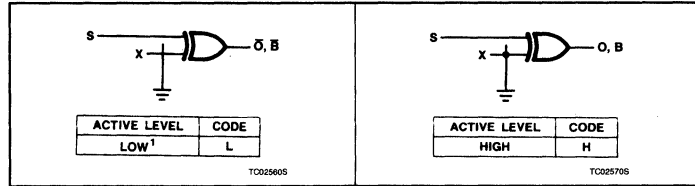
## LOGIC PROGRAMMING

The FPLA can be programmed by means of Logic programming equipment.

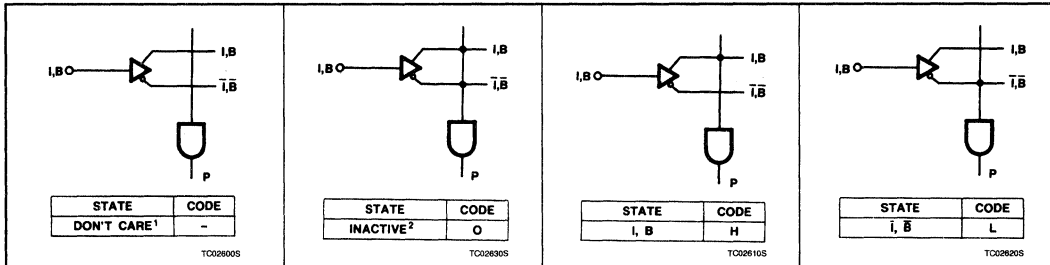
With Logic programming, the AND/OR/EX-OR gate input connections necessary to implement the desired logic function are coded directly from logic equations using the Program Table on the following page.

In this table, the logic state of variables I, P, and B associated with each Sum Term S is assigned a symbol which results in the proper fusing pattern of corresponding links defined as follows:

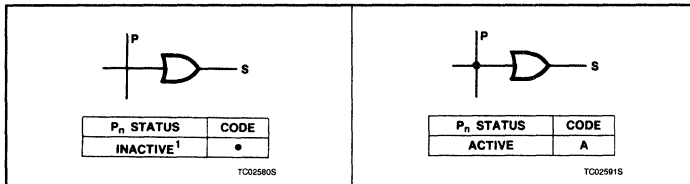
## OUTPUT POLARITY - (O, B)



## "AND" ARRAY - (I, B)



## OR ARRAY - (O, B)



## VIRGIN STATE

A factory shipped virgin device contains all fusible links intact, such that:

1. All outputs at "L" polarity.
2. All P<sub>n</sub> terms are enabled. (Don't Cares).
3. All P<sub>n</sub> terms are inactive on all outputs.

## NOTES:

1. This is the initial unprogrammed state of all links.
2. Any gate P<sub>n</sub> will be unconditionally inhibited if the true and complement of either input (I or B) are both programmed for a connection.

5

# Field-Programmable Logic Array (20 × 24 × 11)

PLHS473

## FPLA PROGRAM TABLE

CUSTOMER NAME _____ PURCHASE ORDER # _____ SIGNETICS DEVICE # _____ CF (XXXX) _____ CUSTOMER SYMBOLIZED PART # _____ TOTAL NUMBER OF PARTS _____ PROGRAM TABLE # _____ REV _____ DATE _____																				<b>AND</b> <table border="1" style="width: 100%; text-align: center;"> <tr> <td style="padding: 2px;">INACTIVE</td> <td style="padding: 2px;">0</td> </tr> <tr> <td style="padding: 2px;">I, B</td> <td style="padding: 2px;">H</td> </tr> <tr> <td style="padding: 2px;">I, B</td> <td style="padding: 2px;">L</td> </tr> <tr> <td style="padding: 2px;">DONT CARE</td> <td style="padding: 2px;">---</td> </tr> </table> I,B(I)										INACTIVE	0	I, B	H	I, B	L	DONT CARE	---	<b>OR</b> <table border="1" style="width: 100%; text-align: center;"> <tr> <td style="padding: 2px;">ACTIVE</td> <td style="padding: 2px;">A</td> <td rowspan="2" style="padding: 2px;">B(O)</td> </tr> <tr> <td style="padding: 2px;">INACTIVE</td> <td style="padding: 2px;">•</td> </tr> </table> <b>CONTROL</b> <table border="1" style="width: 100%; text-align: center;"> <tr> <td style="padding: 2px;">HIGH</td> <td style="padding: 2px;">H</td> <td rowspan="2" style="padding: 2px;">(POL)</td> </tr> <tr> <td style="padding: 2px;">LOW</td> <td style="padding: 2px;">L</td> </tr> </table>										ACTIVE	A	B(O)	INACTIVE	•	HIGH	H	(POL)	LOW	L	<b>POLARITY</b> <table border="1" style="width: 100%; text-align: center;"> <tr> <td style="width: 10px; height: 15px;"></td><td style="width: 10px; height: 15px;"></td><td style="width: 10px; height: 15px;"></td><td style="width: 10px; height: 15px;"></td><td style="width: 10px; height: 15px;"></td><td style="width: 10px; height: 15px;"></td><td style="width: 10px; height: 15px;"></td><td style="width: 10px; height: 15px;"></td><td style="width: 10px; height: 15px;"></td><td style="width: 10px; height: 15px;"></td><td style="width: 10px; height: 15px;"></td><td style="width: 10px; height: 15px;"></td><td style="width: 10px; height: 15px;"></td><td style="width: 10px; height: 15px;"></td><td style="width: 10px; height: 15px;"></td><td style="width: 10px; height: 15px;"></td><td style="width: 10px; height: 15px;"></td><td style="width: 10px; height: 15px;"></td><td style="width: 10px; height: 15px;"></td><td style="width: 10px; height: 15px;"></td> </tr> </table>																													
INACTIVE	0																																																																																						
I, B	H																																																																																						
I, B	L																																																																																						
DONT CARE	---																																																																																						
ACTIVE	A	B(O)																																																																																					
INACTIVE	•																																																																																						
HIGH	H	(POL)																																																																																					
LOW	L																																																																																						
<b>T E R M</b>	<b>AND</b>																				<b>T E R M</b>	<b>OR</b>																																																																	
	<b>I</b>										<b>B (I)</b>											<b>E</b>					<b>D</b>					<b>O</b>					<b>B (O)</b>																																																		
	10	9	8	7	6	5	4	3	2	1	0	8	7	6	5	4	3	2	1	0		B	A	8	7	6	5	4	3	2	1	0	B	A	8	7	6	5	4	3	2	1	0	19	18	23	22	21	20	17	16	15	14	13																																	
	0																																																																																						
	1																																																																																						
	2																																																																																						
	3																																																																																						
	4																																																																																						
	5																																																																																						
	6																																																																																						
	7																																																																																						
	8																																																																																						
	9																																																																																						
	10																																																																																						
	11																																																																																						
	12																																																																																						
	13																																																																																						
	14																																																																																						
	15																																																																																						
	16																																																																																						
	17																																																																																						
	18																																																																																						
	19																																																																																						
	20																																																																																						
21																																																																																							
22																																																																																							
23																																																																																							
<b>PIN</b>	11	10	9	8	7	6	5	4	3	2	1	23	22	21	20	17	16	15	14	13												19	18	23	22	21	20	17	16	15	14	13																																													
<b>VARIABLE NAME</b>																																																																																							

- Notes**
1. The FPLA is shipped with all links open.
  2. Unused I and B bits in the AND array exist as Don't Care (-) in the virgin state.
  3. All P-terms are inactive on all outputs (B, O) in the virgin array.
  4. Unused product terms can be left blank.

13041483

## PLC473 Erasable-Programmable Logic Array (20 × 24 × 11)

Signetics Programmable Logic  
*Preliminary Specification*

### Application Specific Products • Series 24

#### DESCRIPTION

The PLC473 is a two-level logic CMOS Erasable Programmable Logic Device (EPLD) consisting of 24 AND gates and 22 OR gates with EPROM cell connections for programming I/O polarity and direction. The Signetics state-of-the-art Floating Gate CMOS process is used to produce performance not yet achieved in devices of this complexity.

All AND gate inputs are linked to 11 dedicated input pins ( $I_0 - I_{10}$ ) and 9 bi-directional I/O pins ( $B_0 - B_8$ ). These bi-directional pins are controlled via the OR array. Employing the 2 dedicated outputs ( $O_A - O_B$ ) and the programmable I/O direction feature, the PLC473 can be configured with up to 20 inputs — and as many as 11 outputs.

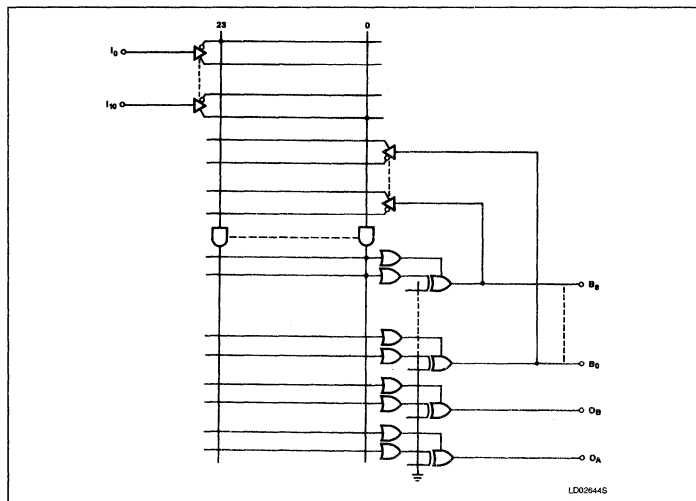
The AND array input buffers provide both the True and Complement of the inputs ( $I_X$ ) and the bidirectional signals ( $B_X$ ) as programmable connections to the AND gates. All 24 AND gates can then be optionally linked to all 22 OR gates (a feature known as Product Term sharing not found in PALs<sup>1</sup> or most

macrocell architectures). The OR array drives 11 output buffers which can be programmed as Active-High for AND-OR functions or Active-Low for AND-NOR functions. In addition, the I/O configuration of each bidirectional pin is individually controlled by a sum-of-products (AND-OR) function which may also contain any of the 24 AND gate outputs. This allows dynamic I/O configuration of all 9 bidirectional pins.

The PLC473 contains two new features of significance. Additional testing capability includes stress test mode, whereby all cells can be stressed to ensure reliability. In another test mode, the threshold voltage of each cell can be individually checked to ensure charge retention.

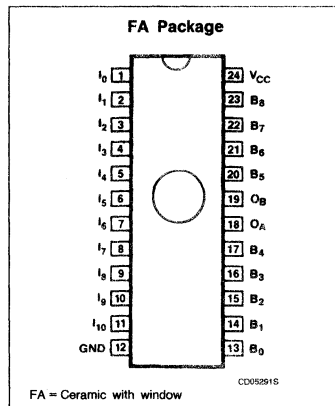
The PLC473 is field programmable using Floating Gate Ultraviolet Erasable Cells. This enables the generation of custom logic patterns using standard programming equipment.

#### FUNCTIONAL DIAGRAM



<sup>1</sup>PAL is a trademark of MMI.

#### PIN CONFIGURATION



#### FEATURES

- Electrically Programmable; UV erasable
- 11 dedicated inputs
- 2 dedicated outputs
- 9 bidirectional I/O lines
- 24 product terms
- I/O direction decoded in OR array
- I/O propagation delay: 35ns (max.)
- Input loading:  $-10\mu A$  (max.)
- Power dissipation at 10MHz CMOS: 210mW  
TTL: 270mW
- Threshold voltage test mode for individual cells
- Stress test mode for individual cells
- Output: Tri-state condition decoded in OR array
- TTL compatible

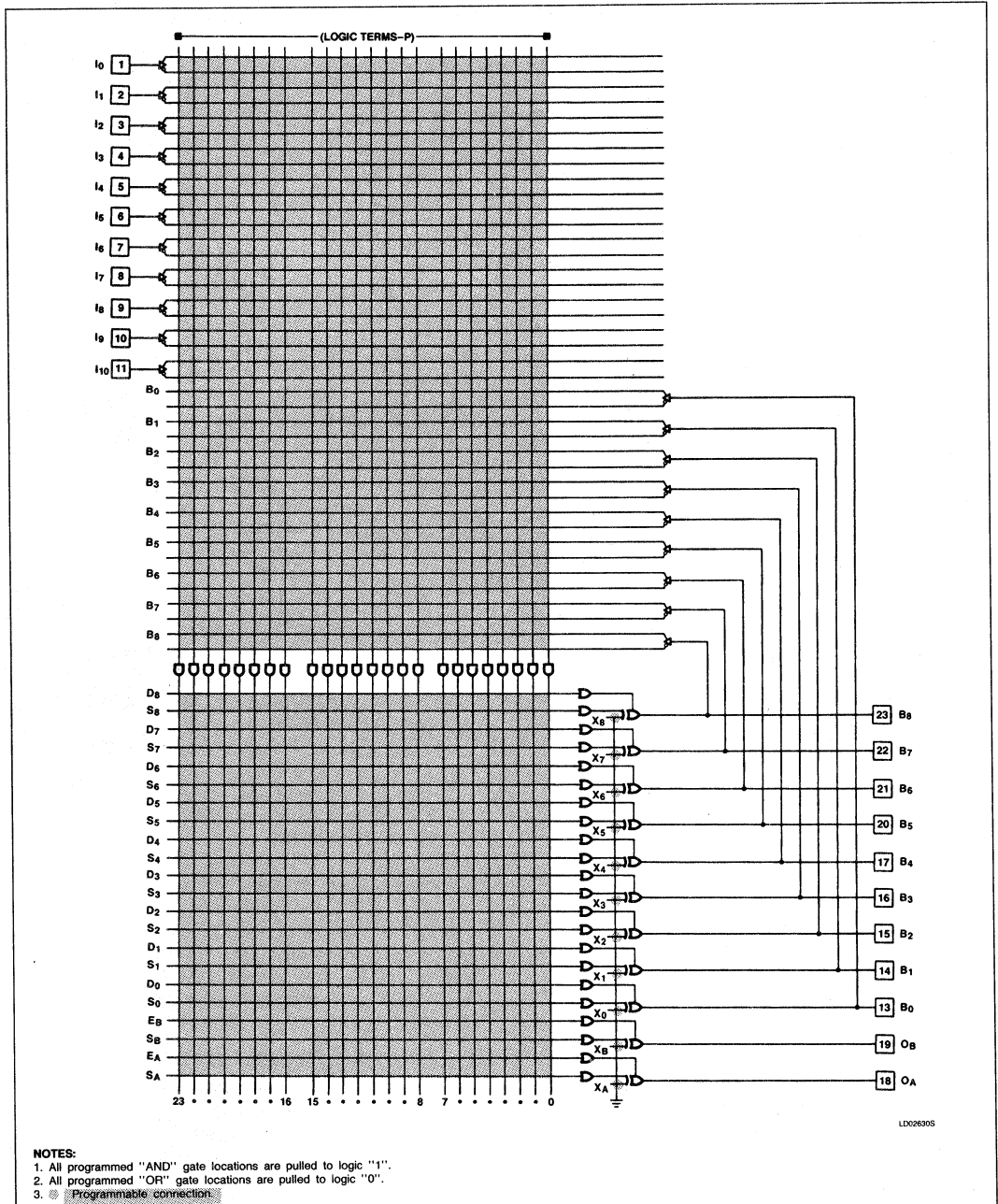
#### APPLICATIONS

- Random logic
- Code converters
- Fault detectors
- Function generators
- Address mapping
- Multiplexing

# Erasable-Programmable Logic Array (20 × 24 × 11)

PLC473

## FPLA LOGIC DIAGRAM



LD026305

## Erasable-Programmable Logic Array (20 X 24 X 11)

PLC473

## ORDERING CODE

DESCRIPTION	ORDER CODE
24-pin Ceramic DIP with Window 300mil-wide	PLC473-35FA
	PLC473-45FA
	PLC473-60FA

## LOGIC FUNCTION

**TYPICAL PRODUCT TERM:**  
 $P_n = A \cdot B \cdot C \cdot D \cdot \dots$

**TYPICAL LOGIC FUNCTION:**  
**AT OUTPUT POLARITY = H**  
 $Z = P_0 + P_1 + P_2 \dots$

**AT OUTPUT POLARITY = L**  
 $Z = P_0 \cdot P_1 \cdot P_2 \dots$

## NOTES:

- For each of the 11 outputs, either function Z (Active-High) or  $\bar{Z}$  (Active-Low) is available, but not both. The desired output polarity is programmed via the EX-OR gates.
- Z, A, B, C, etc. are user defined connections to fixed inputs (I), fixed output pins (O) and bidirectional pins (B).

ABSOLUTE MAXIMUM RATINGS<sup>1</sup>

SYMBOL	PARAMETER	RATING		UNIT
		Min	Max	
$V_{CC}$	Supply voltage		+7	$V_{DC}$
$V_{IN}$	Input voltage		$V_{CC} + 0.5$	$V_{DC}$
$V_{OUT}$	Output voltage		$V_{CC} + 0.5$	$V_{DC}$
$I_{IN}$	Input currents	-30	+30	mA
$I_{OUT}$	Output currents		+100	mA
$T_A$ $T_{STG}$	Temperature range Operating Storage	0 -65	+75 +150	°C

DC ELECTRICAL CHARACTERISTICS  $0^\circ\text{C} \leq T_A \leq +75^\circ\text{C}$ ,  $4.75 \leq V_{CC} \leq 5.25\text{V}$ 

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ <sup>2</sup>	Max	
<b>Input voltage<sup>3</sup></b>						
$V_{IL}$	Low	$V_{CC} = \text{Min}$	-0.3		0.8	V
$V_{IH}$	High	$V_{CC} = \text{Max}$	2.0		$V_{CC} + 0.3$	
<b>Output voltage<sup>3</sup></b>						
$V_{OL}$	Low	$V_{CC} = \text{Min}$ $I_{OL} = 8\text{mA}$			0.45	V
$V_{OH}$	High	$I_{OH} = -3\text{mA}$	2.4			
<b>Input current</b>						
$I_{IL}$	Low	$V_{IN} = \text{GND}$			-10	$\mu\text{A}$
$I_{IH}$	High	$V_{IN} = V_{CC}$			10	
<b>Output current</b>						
$I_{O(OFF)}$	Hi-Z state <sup>8</sup>	$V_{OUT} = V_{CC}$ $V_{OUT} = \text{GND}$			10	$\mu\text{A}$
$I_{OS}$	Short circuit <sup>4,7</sup>	$V_{OUT} = \text{GND}$	-15		-70	
$I_{CC}$	$V_{CC}$ supply current (Active) <sup>5,6</sup>	No load $f = 1\text{MHz}$	CMOS inputs		40	mA
			TTL inputs		50	
<b>Capacitance</b>						
$I_{IN}$ $C_B$	Input I/O	$V_{CC} = 5\text{V}$ $V_{IN} = 2.0\text{V}$ $V_B = 2.0\text{V}$		6 12		pF

AC ELECTRICAL CHARACTERISTICS  $0^\circ\text{C} \leq T_A \leq +75^\circ\text{C}$ ,  $4.75 \leq V_{CC} \leq 5.25\text{V}$ ,  $R_1 = 470\Omega$ ,  $R_2 = 1\text{k}\Omega$ 

SYMBOL	PARAMETER	TO	FROM	TEST CONDITIONS	PLC473-35		PLC473-45		PLC473-60		UNIT
					MIN	MAX	MIN	MAX	MIN	MAX	
$t_{PD}$	Propagation delay	Output±	Input±	$C_L = 30\text{pF}$		35		45		60	ns
$t_{OE}$	Output enable	Output-	Input±	$C_L = 30\text{pF}$		35		45		60	ns
$t_{OD}$	Output disable <sup>9</sup>	Output+	Input±	$C_L = 30\text{pF}$		35		45		60	ns

Notes on following page.



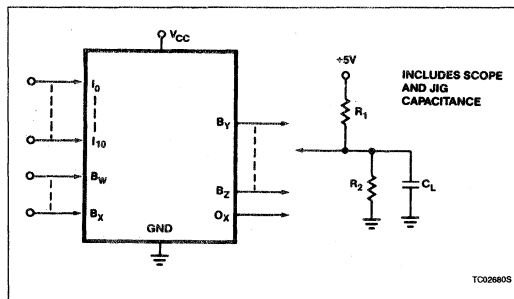
# Erasable-Programmable Logic Array (20 × 24 × 11)

PLC473

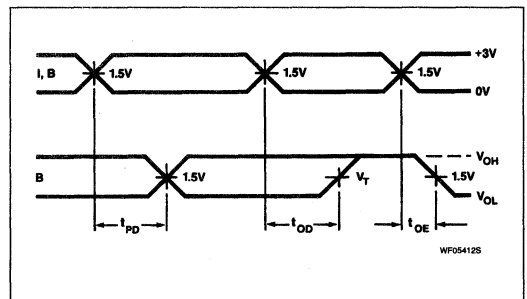
**NOTES:**

1. Stresses above those listed under "Absolute Maximum Ratings" may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.
2. All typical values are at  $V_{CC} = 5V$ ,  $T_A = +25^\circ C$ .
3. All voltage values are with respect to network ground terminal.
4. Test one at a time.
5. CMOS inputs:  $V_{IL} = GND$ ,  $V_{IH} = V_{CC}$   
TTL inputs:  $V_{IL} = 0.45V$ ,  $V_{IH} = 2.4V$
6. Measured with all inputs and outputs switching.
7. Duration of short circuit should not exceed 1 second.
8. Leakage values are a combination of input and output leakage.
9. Measured at  $V_T = V_{OL} + 0.5V$ .

**TEST LOAD CIRCUIT**



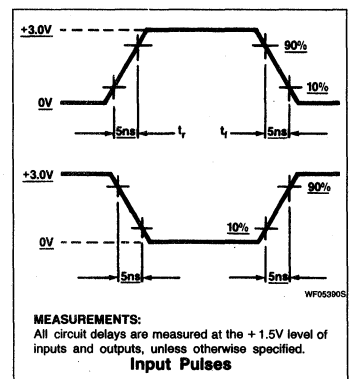
**TIMING DIAGRAM**



**TIMING DEFINITIONS**

- $t_{PD}$  Propagation delay between input and output.
- $t_{OD}$  Delay between input change and when output is off (Hi-Z or High).
- $t_{OE}$  Delay between input change and when output reflects specified output level.

**VOLTAGE WAVEFORM**



# Erasable-Programmable Logic Array (20 × 24 × 11)

PLC473

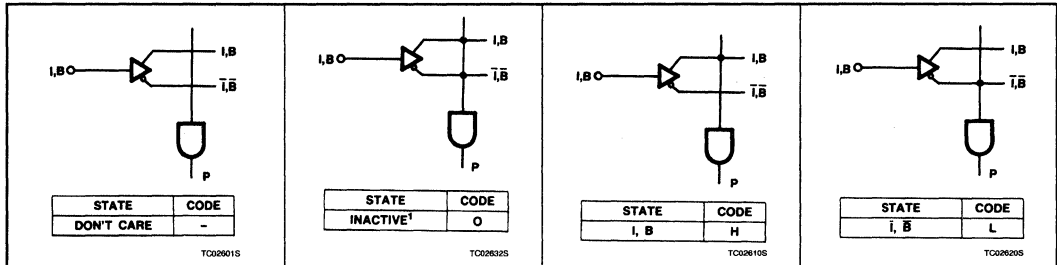
## LOGIC PROGRAMMING

The FPLA can be programmed by means of Logic programming equipment.

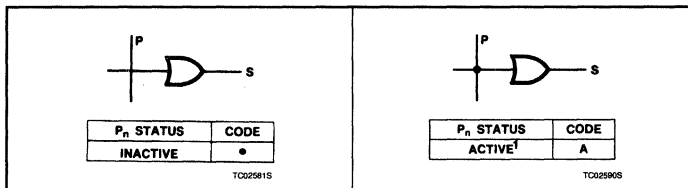
With Logic programming, the AND/OR/EX-OR gate input connections necessary to implement the desired logic function are coded directly from logic equations using the Program Table on the following page.

In this table, the logic state of variables I, P, and B associated with each Sum Term S is assigned a symbol which results in the proper fusing pattern of corresponding link pairs, defined as follows:

### "AND" ARRAY - (I, B)



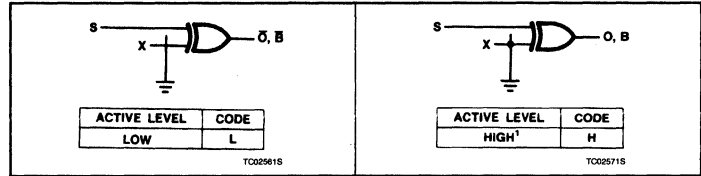
### OR ARRAY - (O, B)



#### NOTES:

1. This is the initial unprogrammed state of all link pairs.
2. Any Product Term (P<sub>n</sub>) will always be False (logic low) if at least one of its (I,B) link pairs is unprogrammed as shown.

### OUTPUT POLARITY - (O, B)



### VIRGIN STATE

A factory shipped virgin device contains all fusible links intact, such that:

1. All outputs at "H" polarity.
2. All P<sub>n</sub> terms are disabled.
3. All P<sub>n</sub> terms are active on all outputs.

5

# Erascble-Programmable Logic Array (20 X 24 X 11)

PLC473

## FPLA PROGRAM TABLE

CUSTOMER NAME _____ PURCHASE ORDER # _____ SIGNETICS DEVICE # _____ CF (XXXX) _____ CUSTOMER SYMBOLIZED PART # _____ TOTAL NUMBER OF PARTS _____ PROGRAM TABLE # _____ REV _____ DATE _____		AND										OR																																																	
		INACTIVE 0 I, B H I, B L DON'T CARE -					I, B(I)					ACTIVE A INACTIVE • B(O)					CONTROL					HIGH H (POL) LOW L					POLARITY																																		
		I										B (I)										E										D										O										B (O)									
		10	9	8	7	6	5	4	3	2	1	0	8	7	6	5	4	3	2	1	0	B	A	8	7	6	5	4	3	2	1	0	B	A	8	7	6	5	4	3	2	1	0	B	A	8	7	6	5	4	3	2	1	0							
0											0										0											0											0																		
1											1										1											1											1																		
2											2										2											2											2																		
3											3										3											3											3																		
4											4										4											4											4																		
5											5										5											5											5																		
6											6										6											6											6																		
7											7										7											7											7																		
8											8										8											8											8																		
9											9										9											9											9																		
10											10										10											10											10																		
11											11										11											11											11																		
12											12										12											12											12																		
13											13										13											13											13																		
14											14										14											14											14																		
15											15										15											15											15																		
16											16										16											16											16																		
17											17										17											17											17																		
18											18										18											18											18																		
19											19										19											19											19																		
20											20										20											20											20																		
21											21										21											21											21																		
22											22										22											22											22																		
23											23										23											23											23																		
PIN	11	10	9	8	7	6	5	4	3	2	1	23	22	21	20	17	16	15	14	13	19	18	23	22	21	20	17	16	15	14	13																														
VARIABLE NAME																																																													

- Notes
1. The FPLA is shipped with all links open.
  2. Unused I and B bits in the AND array exist as Don't Care (-) in the virgin state.
  3. All P-terms are inactive on all outputs (B, O) in the virgin array.
  4. Unused product terms can be left blank.

18010005

### INDEX

#### Section 6 - PLD Data Sheets

##### Series 28

PLS100/PLS101	Field-Programmable Logic Array (16 × 48 × 8) .....	6-3
PLS103	Field-Programmable Gate Array (16 × 9 × 9) .....	6-10
PLS105	Field-Programmable Logic Sequencer (16 × 48 × 8) ...	6-16
PLS105A	Field-Programmable Logic Sequencer (16 × 48 × 8) ...	6-27
PLUS405A	Field-Programmable Logic Sequencer (16 × 64 × 8) ...	6-38



## PLS100/PLS101 Field-Programmable Logic Array (16 × 48 × 8)

Signetics Programmable Logic  
Product Specification

### Application Specific Products • Series 28

#### DESCRIPTION

The PLS100 (Tri-state) and PLS101 (Open Collector) are bipolar, fuse Programmable Logic Arrays (FPLAs). Each device utilizes the standard AND/OR/Invert architecture to directly implement custom sum of product logic equations.

Each device consists of 16 dedicated inputs and 8 dedicated outputs. Each output is capable of being actively controlled by any or all of the 48 product terms. The True, Complement, or Don't Care condition of each of the 16 inputs can be ANDed together to comprise one P-term. All 48 P-terms can be selectively ORed to each output.

The PLS100 and PLS101 are fully TTL compatible, and chip enable control for expansion of input variables and output inhibit. They feature either Open Collector or Tri-state outputs for ease of expansion of product terms and application in bus-organized systems.

Order codes are contained on the pages following.

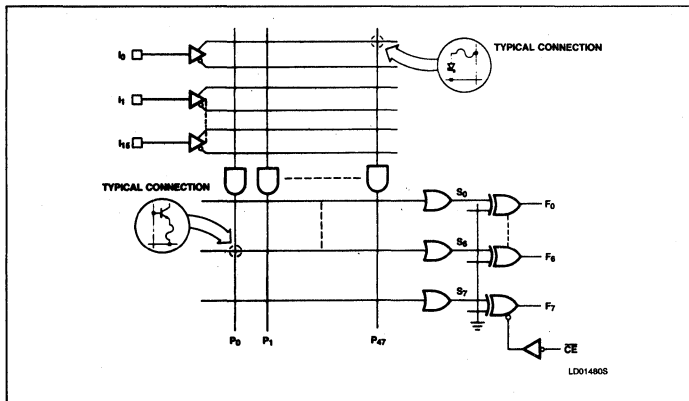
#### FEATURES

- Field-Programmable (Ni-Cr link)
- Input variables: 16
- Output functions: 8
- Product terms: 48
- I/O propagation delay: 50ns (max.)
- Power dissipation: 600mW (typ.)
- Input loading: -100µA (max.)
- Chip Enable input
- Output option:
  - PLS100: Tri-state
  - PLS101: Open-Collector
- Output disable function:
  - Tri-state: Hi-Z
  - Open-Collector: High

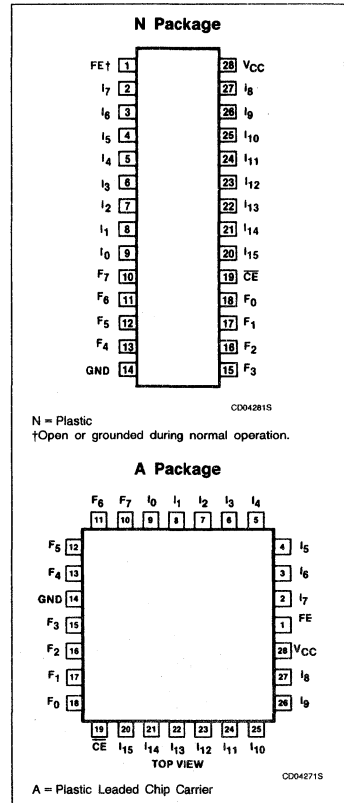
#### APPLICATIONS

- CRT display systems
- Code conversion
- Peripheral controllers
- Function generators
- Look-up and decision tables
- Microprogramming
- Address mapping
- Character generators
- Data security encoders
- Fault detectors
- Frequency synthesizers
- 16-bit to 8-bit bus interface
- Random logic replacement

#### FUNCTIONAL DIAGRAM



#### PIN CONFIGURATIONS



#### LOGIC FUNCTION

TYPICAL PRODUCT TERM:

$$P_n = A \cdot B \cdot C \cdot D \cdot \dots$$

TYPICAL LOGIC FUNCTION:

$$\text{AT OUTPUT POLARITY} = H \\ Z = P_0 + P_1 + P_2 \dots$$

AT OUTPUT POLARITY = L

$$Z = P_0 + P_1 + P_2 + \dots \\ Z = P_0 \cdot P_1 \cdot P_2 \dots$$

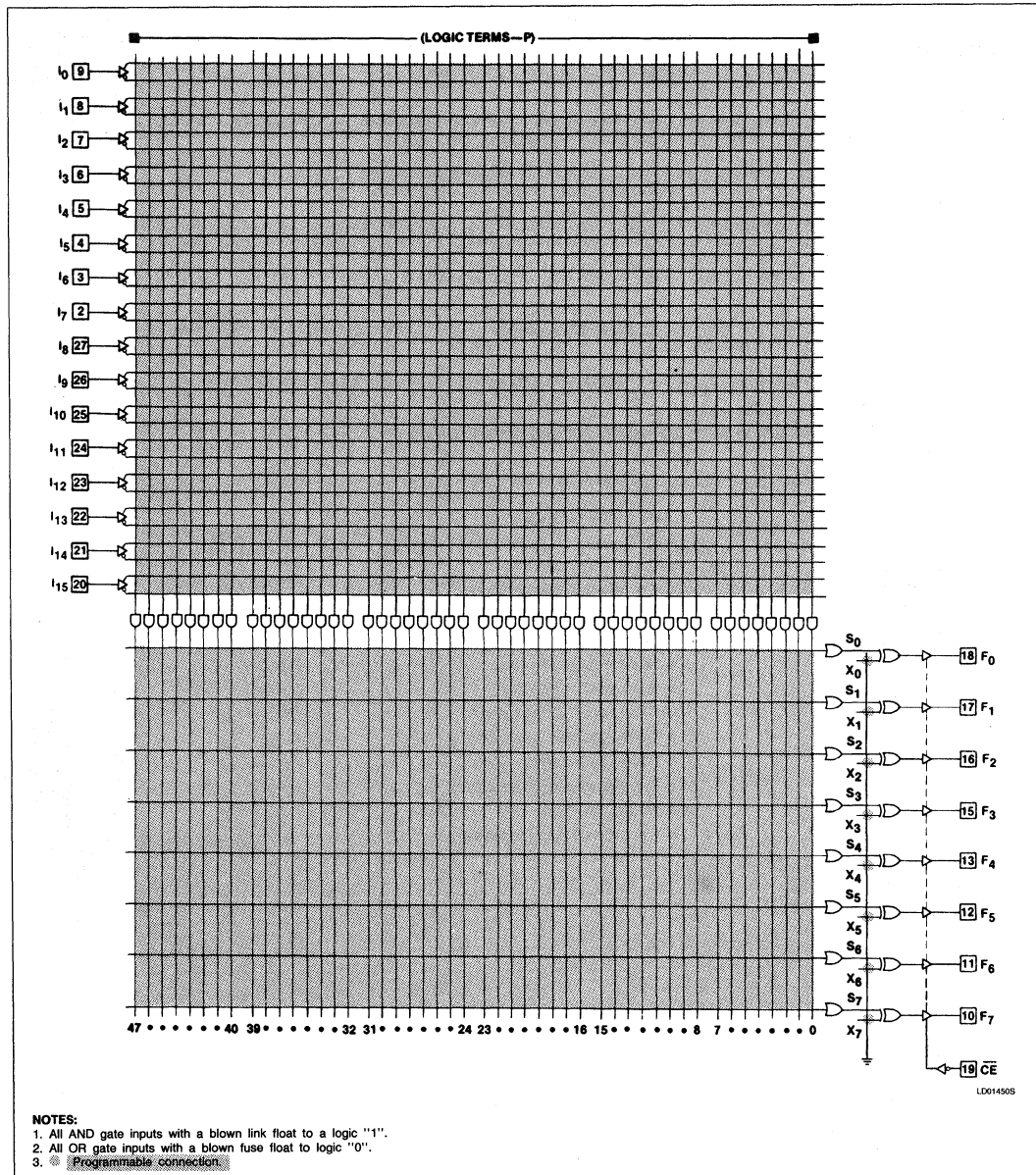
#### NOTES:

1. For each of the 8 outputs, either function Z (Active-High) or Z (Active-Low) is available, but not both. The desired output polarity is programmed via the EX-OR gates.
2. Z, A, B, C, etc. are user defined connections to fixed inputs (I) and output pins (O).

# Field-Programmable Logic Array (16 × 48 × 8)

PLS100/PLS101

## FPLA LOGIC DIAGRAM



# Field-Programmable Logic Array (16 × 48 × 8)

PLS100/PLS101

### ORDERING INFORMATION

DESCRIPTION	TRI-STATE	OPEN-COLLECTOR
28-pin Plastic DIP 600mil-wide	PLS100N	PLS101N
28-pin Plastic Leaded Chip Carrier	PLS100A	PLS101A

### THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

### ABSOLUTE MAXIMUM RATINGS<sup>1</sup>

SYMBOL	PARAMETER	RATING		UNIT
		Min	Max	
V <sub>CC</sub>	Supply voltage		+7	V <sub>DC</sub>
V <sub>IN</sub>	Input voltage		+5.5	V <sub>DC</sub>
V <sub>OUT</sub>	Output voltage		+5.5	V <sub>DC</sub>
I <sub>IN</sub>	Input currents	-30	+30	mA
I <sub>OUT</sub>	Output currents		+100	mA
T <sub>A</sub> T <sub>STG</sub>	Temperature range Operating Storage	0 -65	+75 +150	°C

The PLS100 device is also processed to military requirements for operation over the military temperature range. For specifications and ordering information consult the Signetics Military Data Book.

### DC ELECTRICAL CHARACTERISTICS 0°C ≤ T<sub>A</sub> ≤ +75°C, 4.75V ≤ V<sub>CC</sub> ≤ 5.25V

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ <sup>2</sup>	Max	
<b>Input voltage<sup>3</sup></b>						
V <sub>IH</sub> V <sub>IL</sub> V <sub>IC</sub>	High Low Clamp <sup>3,4</sup>	V <sub>CC</sub> = Max V <sub>CC</sub> = Min V <sub>CC</sub> = Min, I <sub>IN</sub> = -12mA	2		0.8 -1.2	V
<b>Output voltage<sup>3</sup></b>						
V <sub>OH</sub> V <sub>OL</sub>	High (PLS100) <sup>5</sup> Low <sup>3</sup>	V <sub>CC</sub> = Min I <sub>OH</sub> = -2mA I <sub>OL</sub> = 9.6mA	2.4	0.35	0.45	V
<b>Input current</b>						
I <sub>IH</sub> I <sub>IL</sub>	High Low	V <sub>IN</sub> = 5.5V V <sub>IN</sub> = 0.45V		< 1 -10	25 -100	μA
<b>Output current</b>						
I <sub>O(OFF)</sub>	Hi-Z state (PLS100)	$\overline{CE}$ = High, V <sub>CC</sub> = Max V <sub>OUT</sub> = 5.5V		1	40	μA
I <sub>OS</sub>	Short circuit (PLS100) <sup>4,7</sup>	V <sub>OUT</sub> = 0.45V $\overline{CE}$ = Low, V <sub>OUT</sub> = 0V	-15	-1	-40 -70	mA
I <sub>CC</sub>	V <sub>CC</sub> supply current <sup>8</sup>	V <sub>CC</sub> = Max		120	170	mA
<b>Capacitance</b>						
C <sub>IN</sub> C <sub>OUT</sub>	Input Output	$\overline{CE}$ = High, V <sub>CC</sub> = 5.0V V <sub>IN</sub> = 2.0V V <sub>OUT</sub> = 2.0V		8	17	pF

Notes on following page.

6



# Field-Programmable Logic Array (16 × 48 × 8)

PLS100/PLS101

**AC ELECTRICAL CHARACTERISTICS**  $R_1 = 470\Omega$ ,  $R_2 = 1k\Omega$ ,  $C_L = 30pF$ ,  $0^\circ C \leq T_A \leq +75^\circ C$ ,  $4.75V \leq V_{CC} \leq 5.25V$

SYMBOL	PARAMETER	TO	FROM	LIMITS			UNIT
				Min	Typ <sup>2</sup>	Max	
<b>Propagation delay</b>							
$t_{PD}$	Input	Output	Input		35	50	ns
$t_{CE}$	Chip enable	Output	Chip enable		15	30	ns
<b>Disable time</b>							
$t_{CD}$	Chip disable	Output	Chip enable		15	30	ns

**NOTES:**

- Stresses above those listed under "Absolute Maximum Ratings" may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other conditions above those indicated in the operational and programming specification of the device is not implied.
- All values are at  $V_{CC} = 5V$ ,  $T_A = +25^\circ C$ .
- All voltage values are with respect to network ground terminal.
- Test one pin at a time.
- Measured with  $V_{IL}$  applied to  $\overline{CE}$  and a logic high stored.
- Measured with a programmed logic condition for which the output test is at a low logic level. Output sink current is applied through a resistor to  $V_{CC}$ .
- Duration of short circuit should not exceed 1 second.
- $I_{CC}$  is measured with the chip enable input grounded, all other inputs at 4.5V and the outputs open.

# Field-Programmable Logic Array (16 × 48 × 8)

PLS100/PLS101

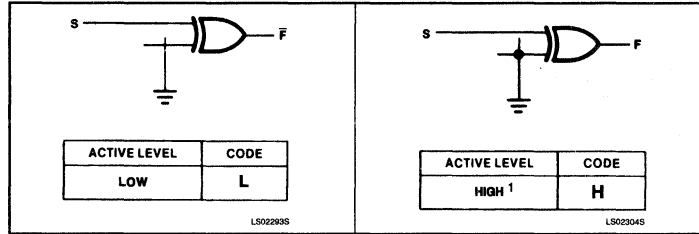
## LOGIC PROGRAMMING

The FPLA can be programmed by means of Logic programming equipment.

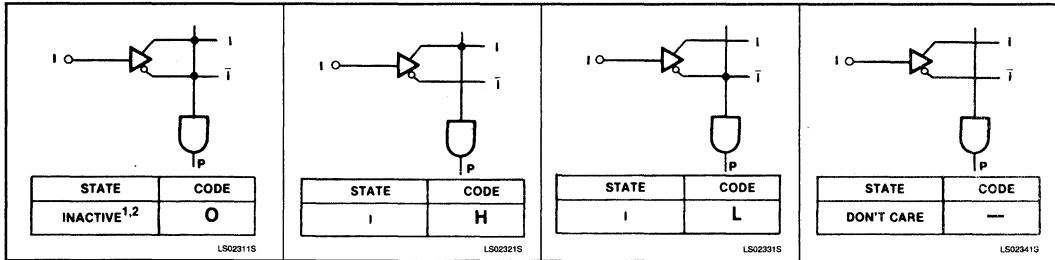
With Logic programming, the AND/OR/EX-OR gate input connections necessary to implement the desired logic function are coded directly from logic equations using the Program Table.

In this Table the logic state or action of variables I, P, and F, associated with each Sum Term  $S_n$ , is assigned a symbol which results in the proper fusing pattern of corresponding link pairs, defined as follows:

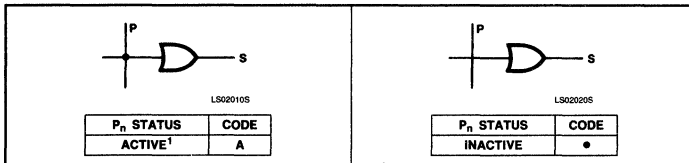
## OUTPUT POLARITY - (F)



## "AND" ARRAY - (I)



## "OR" ARRAY - (F)



### NOTES:

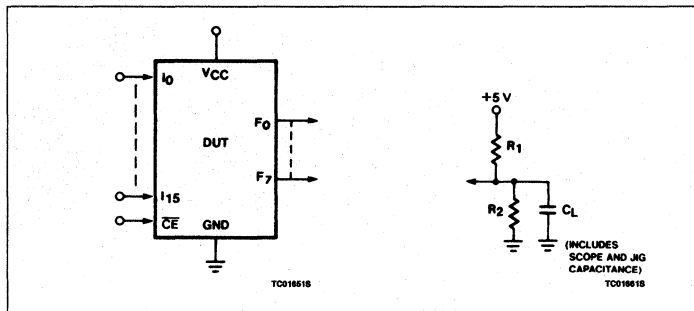
1. This is the initial unprogrammed state of all link pairs. It is normally associated with all unused (inactive) AND gates  $P_n$ .
2. Any gate  $P_n$  will be unconditionally inhibited if any one of its (I) link pairs is left intact.

6

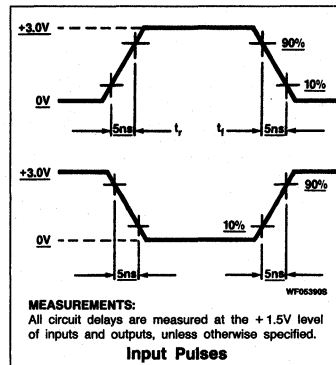
# Field-Programmable Logic Array (16 × 48 × 8)

PLS100/PLS101

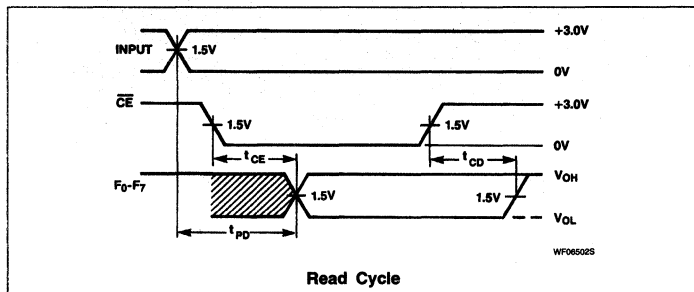
### TEST LOAD CIRCUIT



### VOLTAGE WAVEFORM



### TIMING DIAGRAM



### TIMING DEFINITIONS

- $t_{CE}$  Delay between beginning of Chip Enable low (with Input valid) and when Data Output becomes valid.
- $t_{CD}$  Delay between when Chip Enable becomes High and Data Output is in off state (Hi-Z or High).
- $t_{PD}$  Delay between beginning of valid Input (with Chip Enable Low) and when Data Output becomes valid.

### VIRGIN STATE

The PLS100/101 virgin devices are factory shipped in an unprogrammed state, with all fuses intact, such that:

1. All  $P_n$  terms are disabled (inactive).
2. All  $P_n$  terms are active on all outputs.
3. All outputs are Active-High.

# Field-Programmable Logic Array (16 × 48 × 8)

## PLS100/PLS101

### FPLA PROGRAM TABLE

				PROGRAM TABLE ENTRIES															
				INPUT VARIABLE			OUTPUT FUNCTION		OUTPUT ACTIVE LEVEL										
CUSTOMER NAME _____				Im	Im	Don't Care	Prod. Term Present in F <sub>p</sub>	Prod. Term Not Present in F <sub>p</sub>	Active High	Active Low									
PURCHASE ORDER # _____				H	L	— (dash)	A	• (period)	H	L									
SIGNETICS DEVICE # _____ CF (XXXX)				NOTE Enter (–) for unused inputs of used P-terms.				NOTES 1. Entries independent of output polarity. 2. Enter (A) for unused outputs of used P-terms.		NOTES 1. Polarity programmed once only. 2. Enter (H) for all unused outputs.									
CUSTOMER SYMBOLIZED PART # _____																			
TOTAL NUMBER OF PARTS _____																			
PROGRAM TABLE # _____ REV _____ DATE _____																			
VARIABLE NAME	PIN NO.	AND									POLARITY OUTPUT (F <sub>0</sub> )								
		INPUT (I <sub>m</sub> )																	
	0	1	2	3	4	5	6	7	8	9	7	6	5	4	3	2	1	0	
	1																		
	2																		
	3																		
	4																		
	5																		
	6																		
	7																		
	8																		
	9																		
	10																		
	11																		
	12																		
	13																		
	14																		
	15																		
	16																		
	17																		
	18																		
	19																		
	20																		
	21																		
	22																		
	23																		
	24																		
	25																		
	26																		
	27																		
	28																		
	29																		
	30																		
	31																		
	32																		
	33																		
	34																		
	35																		
	36																		
	37																		
	38																		
	39																		
	40																		
	41																		
	42																		
	43																		
	44																		
	45																		
	46																		
	47																		
	48	1	1	1	1	1	1	1	1	1									
	49	0	1	2	3	5	6	7	8										

## PLS103 Field-Programmable Gate Array (16 × 9 × 9)

Signetics Programmable Logic  
Product Specification

### Application Specific Products • Series 28

#### DESCRIPTION

The PLS103 is a bipolar, fuse Programmable Gate Array. The device consists of nine AND/NAND gates which share 16 common inputs. The type of gate is selected by programming the output as Active-High (H) or Active-Low (L). Each of the 16 inputs  $I_0 - I_{15}$  can be programmed to provide the True (H), Complement (L), or Don't Care (—) state to each of the nine AND/NAND gates. OR/NOR logic functions can also be implemented by complementing the inputs and outputs via on-chip inverting buffers.

The device is field programmable, which means that custom patterns are immediately available.

The PLS103 includes chip-enable control for output strobing and inhibit. It features Tri-state outputs for ease of expansion of input variables and application in bus-organized systems.

Order codes are contained on the pages following.

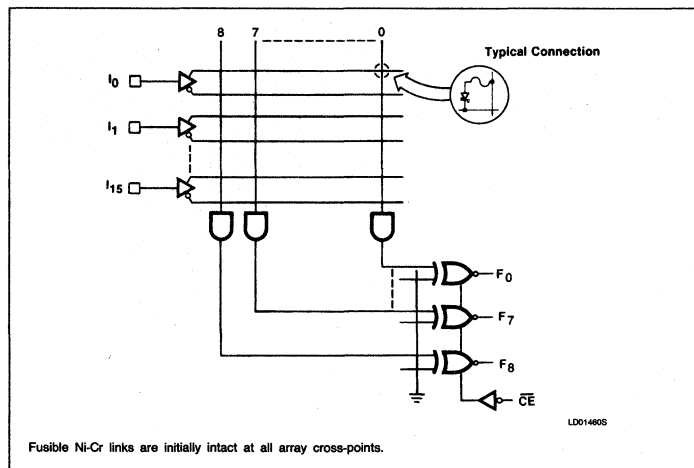
#### FEATURES

- Field-Programmable (Ni-Cr link)
- 16 input variables
- 9 output functions
- Chip Enable input
- I/O propagation delay: 35ns (max.)
- Power dissipation: 600mW (typ.)
- Input loading:  $-100\mu\text{A}$  (max.)
- Tri-state outputs
- Output disable function: HI-Z
- Fully TTL compatible

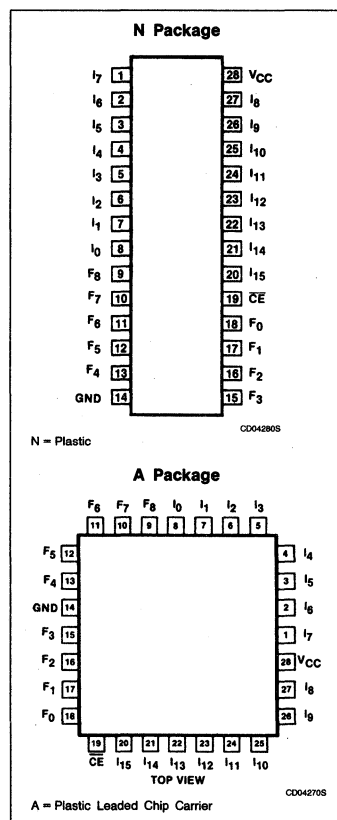
#### APPLICATIONS

- Random logic
- Address decoders
- Code detectors
- Peripheral selectors
- Fault monitors
- Machine state decoders

#### FUNCTIONAL DIAGRAM



#### PIN CONFIGURATIONS



#### LOGIC FUNCTION

TYPICAL OUTPUT FUNCTIONS:  
**ACTIVE-HIGH**

$$X = A \cdot \bar{B} \cdot C \cdot \dots$$

**ACTIVE-LOW**

$$X = \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \dots$$

$$X = \bar{A} + \bar{B} + \bar{C} + \dots$$

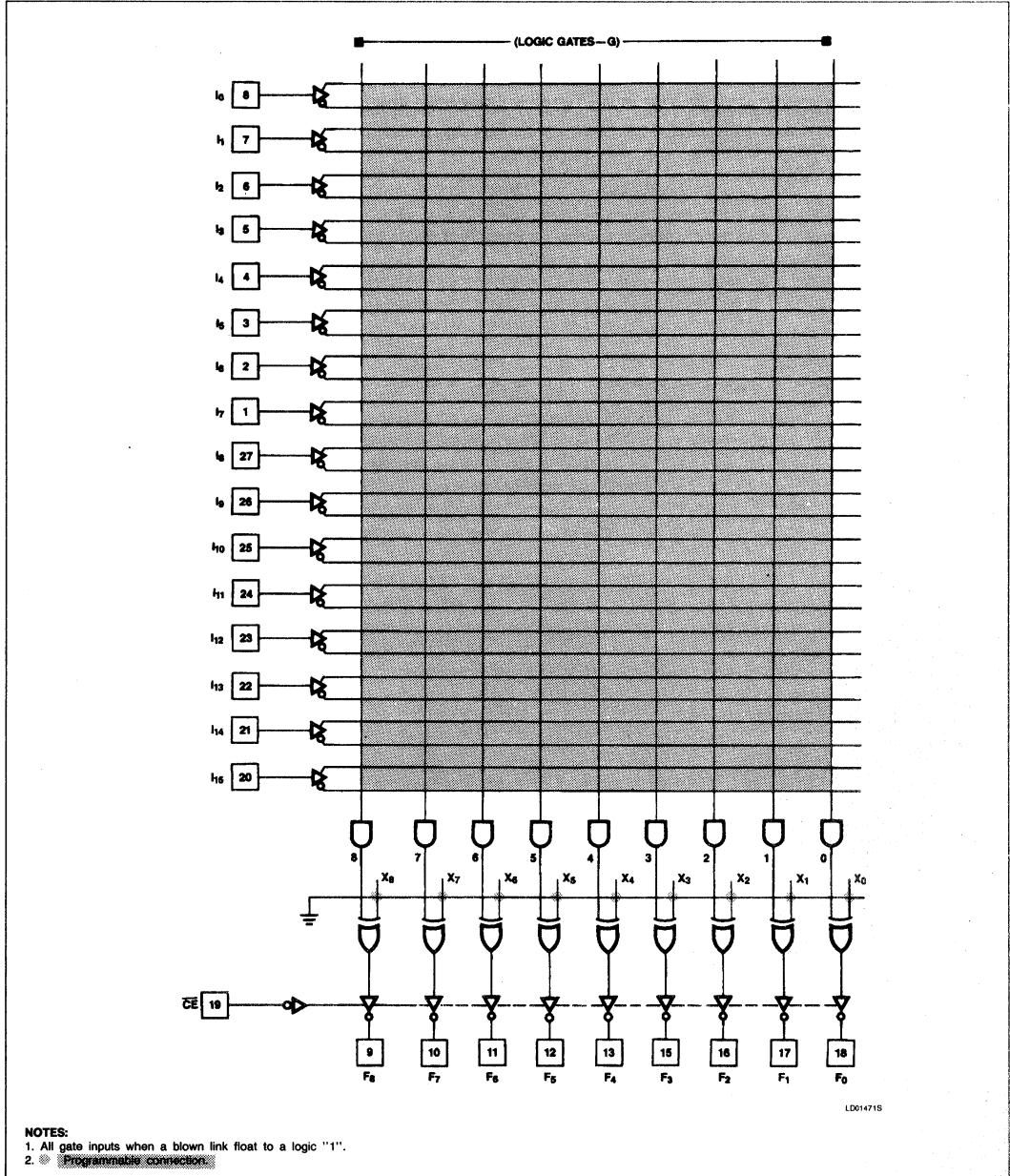
#### NOTES:

1. For each of the 9 outputs, either function X (Active-High) or  $\bar{X}$  (Active-Low) is available, but not both. The desired output polarity is programmed via the EX-OR gates.
2. X, A, B, C, etc. are user defined connections to fixed inputs (I) and output pins (O).

# Field-Programmable Gate Array (16 × 9 × 9)

PLS103

## FPGA LOGIC DIAGRAM



6

# Field-Programmable Gate Array (16 × 9 × 9)

PLS103

**ORDERING INFORMATION**

DESCRIPTION	ORDER CODE
28-pin Plastic DIP 600mil-wide	PLS103N
28-pin Plastic Leaded Chip Carrier	PLS103A

**THERMAL RATINGS**

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

**ABSOLUTE MAXIMUM RATINGS**

SYMBOL	PARAMETER	RATING		UNIT
		Min	Max	
V <sub>CC</sub>	Supply voltage		+7	V <sub>DC</sub>
V <sub>IN</sub>	Input voltage		+5.5	V <sub>DC</sub>
V <sub>OUT</sub>	Output voltage		+5.5	V <sub>DC</sub>
I <sub>IN</sub>	Input current	-30	+30	mA
I <sub>OUT</sub>	Output current		+100	mA
T <sub>A</sub> T <sub>STG</sub>	Temperature range Operating Storage	0 -65	+75 +150	°C

**DC ELECTRICAL CHARACTERISTICS** 0°C ≤ T<sub>A</sub> ≤ +75°C, 4.75V ≤ V<sub>CC</sub> ≤ 5.25V

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ <sup>2</sup>	Max	
<b>Input voltage<sup>1</sup></b>						
V <sub>IH</sub> V <sub>IL</sub> V <sub>IC</sub>	High Low Clamp <sup>3</sup>	V <sub>CC</sub> = Max V <sub>CC</sub> = Min V <sub>CC</sub> = Min, I <sub>IN</sub> = -12mA	2.0		0.8 -1.2	V
<b>Output voltage<sup>1</sup></b>						
V <sub>OH</sub> V <sub>OL</sub>	High <sup>5</sup> Low <sup>4</sup>	V <sub>CC</sub> = Min I <sub>OH</sub> = -2mA I <sub>OL</sub> = 9.6mA	2.4	0.35	0.45	V
<b>Input current</b>						
I <sub>IH</sub> I <sub>IL</sub>	High Low	V <sub>IN</sub> = 5.5V V <sub>IN</sub> = 0.45V		< 1 -10	25 -100	μA
<b>Output current</b>						
I <sub>O(OFF)</sub> I <sub>OS</sub>	Hi-Z state Short circuit <sup>3,6</sup>	$\overline{CE}$ = High, V <sub>CC</sub> = Max V <sub>OUT</sub> = 5.5V V <sub>OUT</sub> = 0.45V $\overline{CE}$ = Low, V <sub>OUT</sub> = 0V		1 -1	40 -40 -70	μA mA
I <sub>CC</sub>	V <sub>CC</sub> supply current <sup>7</sup>	V <sub>CC</sub> = Max		120	170	mA
<b>Capacitance</b>						
C <sub>IN</sub> C <sub>OUT</sub>	Input Output	V <sub>CC</sub> = 5.0V V <sub>IN</sub> = 2.0V V <sub>OUT</sub> = 2.0V		8 15		pF

Notes on following page.

# Field-Programmable Gate Array (16 × 9 × 9)

PLS103

**AC ELECTRICAL CHARACTERISTICS**  $R_1 = 470\Omega$ ,  $R_2 = 1k\Omega$ ,  $C_L = 30pF$ ,  $0^\circ C \leq T_A \leq +75^\circ C$ ,  $4.75V \leq V_{CC} \leq 5.25V$ 

SYMBOL	PARAMETER	TO	FROM	LIMITS			UNIT
				Min	Typ <sup>2</sup>	Max	
<b>Propagation delay</b>							
$t_{PD}$	Input	Output	Input		20	35	ns
$t_{CE}$	Chip enable	Output	Chip enable		15	30	ns
<b>Disable time</b>							
$t_{CD}$	Chip disable	Output	Chip enable		15	30	ns

**NOTES:**

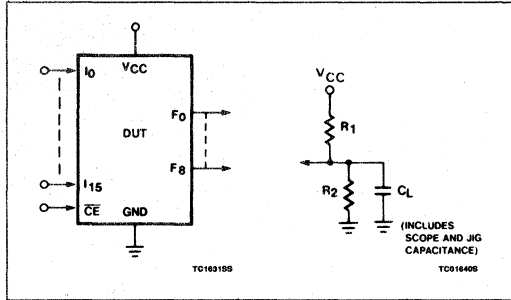
- All voltage values are with respect to network ground terminal.
- All typical values are at  $V_{CC} = 5V$ ,  $T_A = +25^\circ C$ .
- Test one pin at a time.
- Measure with a programmed logic condition for which the output under test is at a low logic level. Output sink current is supplied through a resistor to  $V_{CC}$ .
- Measured with  $V_{IL}$  applied to  $\overline{CE}$  and a logic high at the output.
- Duration of short circuit should not exceed 1 second.
- $t_{CC}$  is measured with the outputs open.
- Stresses above those listed under "Absolute Maximum Ratings" may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other conditions above those listed in the operational and programming specification of the device is not implied.



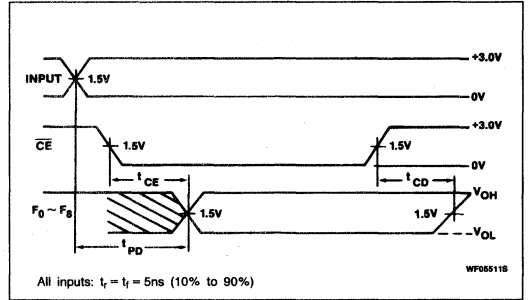
# Field-Programmable Gate Array (16 × 9 × 9)

PLS103

## TEST LOAD CIRCUIT



## VOLTAGE WAVEFORM



## LOGIC PROGRAMMING

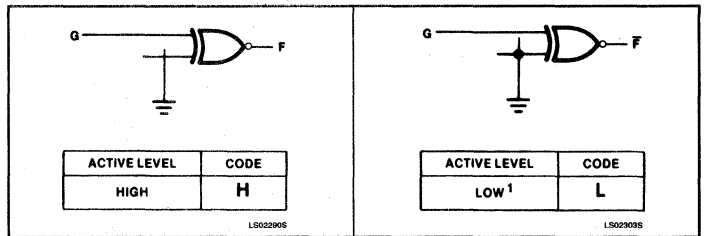
In a virgin device all Ni-Cr links are intact.

The FPGA can be programmed by means of Logic programming equipment.

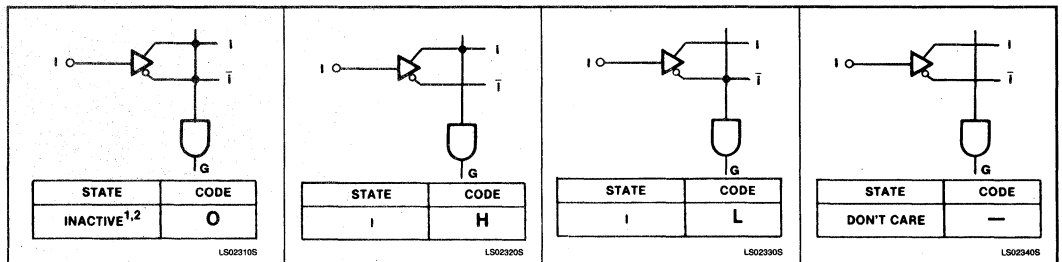
With Logic programming, the AND/EX-OR gate input connections necessary to implement the desired logic function are coded directly from logic equations using the Program Table.

In this table, the logic state of variables I and F associated with each P-term  $P_n$  is assigned a symbol which results in the proper fusing pattern of corresponding link pairs, defined as follows:

## OUTPUT POLARITY - (F)



## "AND" ARRAY - (I), (P)



### NOTES:

1. This is the initial unprogrammed state of all links.
2. Any gate  $G_n$  will be unconditionally inhibited if both the true and complement fuses of any input (I) are left intact.

## VIRGIN STATE

The PLS103 virgin device is factory shipped in an unprogrammed state, with all fuses intact, such that:

1. All  $P_n$  terms are disabled (inactive).
2. All  $P_n$  terms are active on all outputs.
3. All outputs are Active-Low.

# Field-Programmable Gate Array (16 × 9 × 9)

PLS103

## FPGA PROGRAM TABLE

CUSTOMER NAME _____ PURCHASE ORDER # _____ SIGNETICS DEVICE # _____ TOTAL NUMBER OF PARTS _____ PROGRAM TABLE # _____	THIS PORTION TO BE COMPLETED BY SIGNETICS CF (XXXX) _____ CUSTOMER SYMBOLIZED PART # _____ DATE RECEIVED _____ COMMENTS _____
---	---

- F<sub>0</sub> (18) \_\_\_\_\_ = \_\_\_\_\_
- F<sub>1</sub> (17) \_\_\_\_\_ = \_\_\_\_\_
- F<sub>2</sub> (16) \_\_\_\_\_ = \_\_\_\_\_
- F<sub>3</sub> (15) \_\_\_\_\_ = \_\_\_\_\_
- F<sub>4</sub> (13) \_\_\_\_\_ = \_\_\_\_\_
- F<sub>5</sub> (12) \_\_\_\_\_ = \_\_\_\_\_
- F<sub>6</sub> (11) \_\_\_\_\_ = \_\_\_\_\_
- F<sub>7</sub> (10) \_\_\_\_\_ = \_\_\_\_\_
- F<sub>8</sub> (9) \_\_\_\_\_ = \_\_\_\_\_

GATE	INPUT															
POLARITY	I <sub>15</sub>	I <sub>14</sub>	I <sub>13</sub>	I <sub>12</sub>	I <sub>11</sub>	I <sub>10</sub>	I <sub>9</sub>	I <sub>8</sub>	I <sub>7</sub>	I <sub>6</sub>	I <sub>5</sub>	I <sub>4</sub>	I <sub>3</sub>	I <sub>2</sub>	I <sub>1</sub>	I <sub>0</sub>
F <sub>0</sub>																
F <sub>1</sub>																
F <sub>2</sub>																
F <sub>3</sub>																
F <sub>4</sub>																
F <sub>5</sub>																
F <sub>6</sub>																
F <sub>7</sub>																
F <sub>8</sub>																
PIN NO.	20	21	22	23	24	25	26	27	1	2	3	4	5	6	7	8
VARIABLE NAME																

<b>NOTES:</b> 1. The FPGA is shipped with all links intact. Thus a background of entries corresponding to states of virgin links exists in the table, shown BLANK for clarity. 2. Unused inputs are normally programmed Don't Care (—). 3. Unused Gates can be left blank.	<b>PROGRAM TABLE ENTRIES</b>  	<b>AND</b> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr><td>INACTIVE</td><td>0</td></tr> <tr><td>I</td><td>H</td></tr> <tr><td>I</td><td>L</td></tr> <tr><td>Don't Care</td><td>—</td></tr> </table>	INACTIVE	0	I	H	I	L	Don't Care	—	<b>CONTROL</b> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr><td>HIGH</td><td>H</td></tr> <tr><td>LOW</td><td>L</td></tr> </table> (POL.)	HIGH	H	LOW	L
INACTIVE	0														
I	H														
I	L														
Don't Care	—														
HIGH	H														
LOW	L														

6

# PLS105

## Field-Programmable Logic Sequencer (16 × 48 × 8)

Signetics Programmable Logic  
Product Specification

Application Specific Products  
• Series 28

### DESCRIPTION

The PLS105 is a bipolar programmable state machine of the Mealy type. It contains logic AND-OR gate arrays with user programmable connections which control the inputs of on-chip State and Output Registers. These consist respectively of 6  $Q_p$ , and 8  $Q_f$  edge-triggered, clocked S/R flip-flops, with an Asynchronous Preset option. All flip-flops are unconditionally preset to "1" during power turn on.

The AND array combines 16 external inputs  $I_0 - I_{15}$  with six internal inputs  $P_0 - P_5$ , which are fed back from the State Registers to form up to 48 transition terms (AND terms). All transition terms can include True, False, or Don't Care states of the controlling variables, and are merged in the OR array to issue next-state and next-output commands to their respective registers on the Low-to-High transition of the Clock pulse. Both True and Complement transition terms can be generated by optional use of the internal input variable (C) from the Complement Array. Also, if desired, the Preset input can be converted to Output Enable function, as an additional user-programmable option.

Order codes are contained on the pages following.

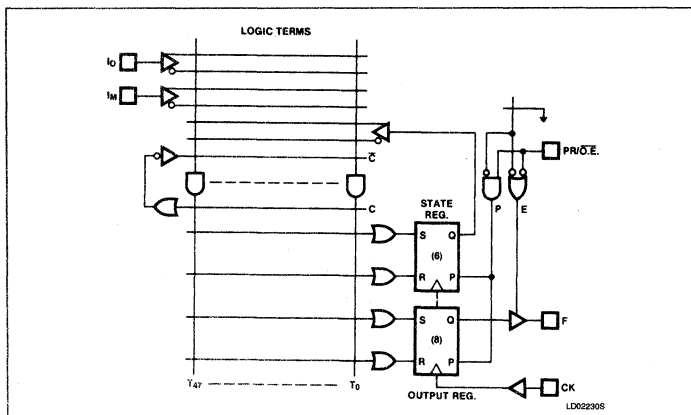
### FEATURES

- Field-Programmable (Ni-Cr link)
- 16 input variables
- 8 output functions
- 48 transition terms
- 6-bit State Register
- 8-bit Output Register
- Transition Complement Array
- Positive edge-triggered clocked flip-flops
- Programmable Asynchronous Preset or Output Enable
- Power-on preset to all "1" of internal registers
- $f_{MAX} = 13.9\text{MHz}$
- Power dissipation: 600mW (typ.)
- TTL compatible
- Single +5V supply
- Tri-state outputs

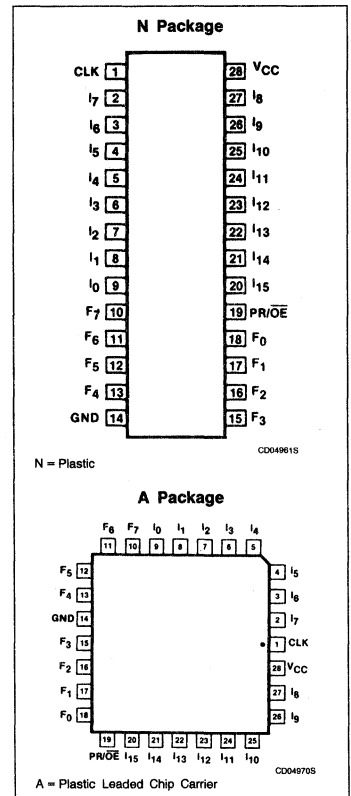
### APPLICATIONS

- Interface protocols
- Sequence detectors
- Peripheral controllers
- Timing generators
- Sequential circuits
- Elevator controllers
- Security locking systems
- Counters
- Shift registers

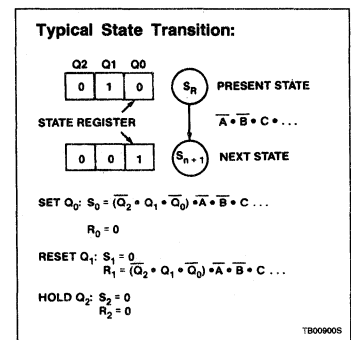
### FUNCTIONAL DIAGRAM



### PIN CONFIGURATIONS



### LOGIC FUNCTION



# Field-Programmable Logic Sequencer (16 × 48 × 8)

PLS105

## PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION	POLARITY
1	CK	<b>Clock:</b> The Clock input to the State and Output Registers. A Low-to-High transition on this line is necessary to update the contents of both registers.	Active-High
2-8 20-27	I <sub>1-15</sub>	<b>Logic Inputs:</b> The 15 external inputs to the AND array used to program jump conditions between machine states, as determined by a given logic sequence.	Active-High/Low
9	I <sub>0</sub>	<b>Logic/Diagnostic Input:</b> A 16th external logic input to the AND array, as above, when exercised with standard TTL levels. When I <sub>0</sub> is held at +10V, device outputs F <sub>0-5</sub> reflect the contents of State Register bits P <sub>0-5</sub> . The contents of each Output Register remains unaltered.	Active-High/Low
10-13 15-18	F <sub>0-7</sub>	<b>Logic/Diagnostic Outputs:</b> Eight device outputs which normally reflect the contents of Output Register bits Q <sub>0-7</sub> , when enabled. When I <sub>0</sub> is held at +10V, F <sub>0-5</sub> = (P <sub>0-5</sub> ), and F <sub>6, 7</sub> = Logic "1".	Active-High
19	PR/ŌE	<b>Preset or Output Enable Input:</b> A user programmable function: <ul style="list-style-type: none"> <li>• <b>Preset:</b> Provides an Asynchronous Preset to logic "1" of all State and Output Register bits. Preset overrides Clock, and when held High, clocking is inhibited and F<sub>0-7</sub> are High. Normal clocking resumes with the first full clock pulse following a High-to-Low clock transition, after Preset goes Low.</li> <li>• <b>Output Enable:</b> Provides an Output Enable function to all output buffers F<sub>0-7</sub> from the Output Register.</li> </ul>	Active-High (H)  Active-Low (L)

## TRUTH TABLE 1, 2, 3, 4, 5, 6

V <sub>CC</sub>	OPTION		I <sub>0</sub>	CK	S	R	Q <sub>P/F</sub>	F	
	PR	ŌE							
+5V	H	█	*	X	X	X	H	H	
	L	█	+10V	X	X	X	Q <sub>n</sub>	(Q <sub>P</sub> ) <sub>n</sub>	
	L	█	X	X	X	X	Q <sub>n</sub>	(Q <sub>F</sub> ) <sub>n</sub>	
	█	H	*	X	X	X	Q <sub>n</sub>	Hi-Z	
	█	L	+10V	X	X	X	Q <sub>n</sub>	(Q <sub>P</sub> ) <sub>n</sub>	
	█	L	X	X	X	X	Q <sub>n</sub>	(Q <sub>F</sub> ) <sub>n</sub>	
	█	L	X	↑	L	L	Q <sub>n</sub>	(Q <sub>F</sub> ) <sub>n</sub>	
	█	L	X	↑	L	H	L	L	
	█	L	X	↑	H	L	H	H	
	█	L	X	↑	H	H	IND.	IND.	
	↑	X	X	X	X	X	X	H	█

### NOTES:

- Positive Logic:  
 $S/R = T_0 + T_1 + T_2 + \dots + T_{47}$   
 $T_n = C(I_0 I_1 I_2 \dots) (P_0 P_1 \dots P_5)$
- Either Preset (Active-High) or Output Enable (Active-Low) are available, but not both. The desired function is a user-programmable option.
- ↑ denotes transition from Low to High level.
- R = S = High is an illegal input condition.
- \* = H/L/+10V
- X = Don't Care (≤5.5V)

## VIRGIN STATE

A factory shipped virgin device contains all fusible links intact, such that:

- PR/ŌE option is set to PR. Thus, all outputs will be at "1", as preset by initial power-up procedure.
- All transition terms are disabled (0).
- All S/R flip-flop inputs are disabled (0).
- The device can be clocked via a Test Array pre-programmed with a standard test pattern.

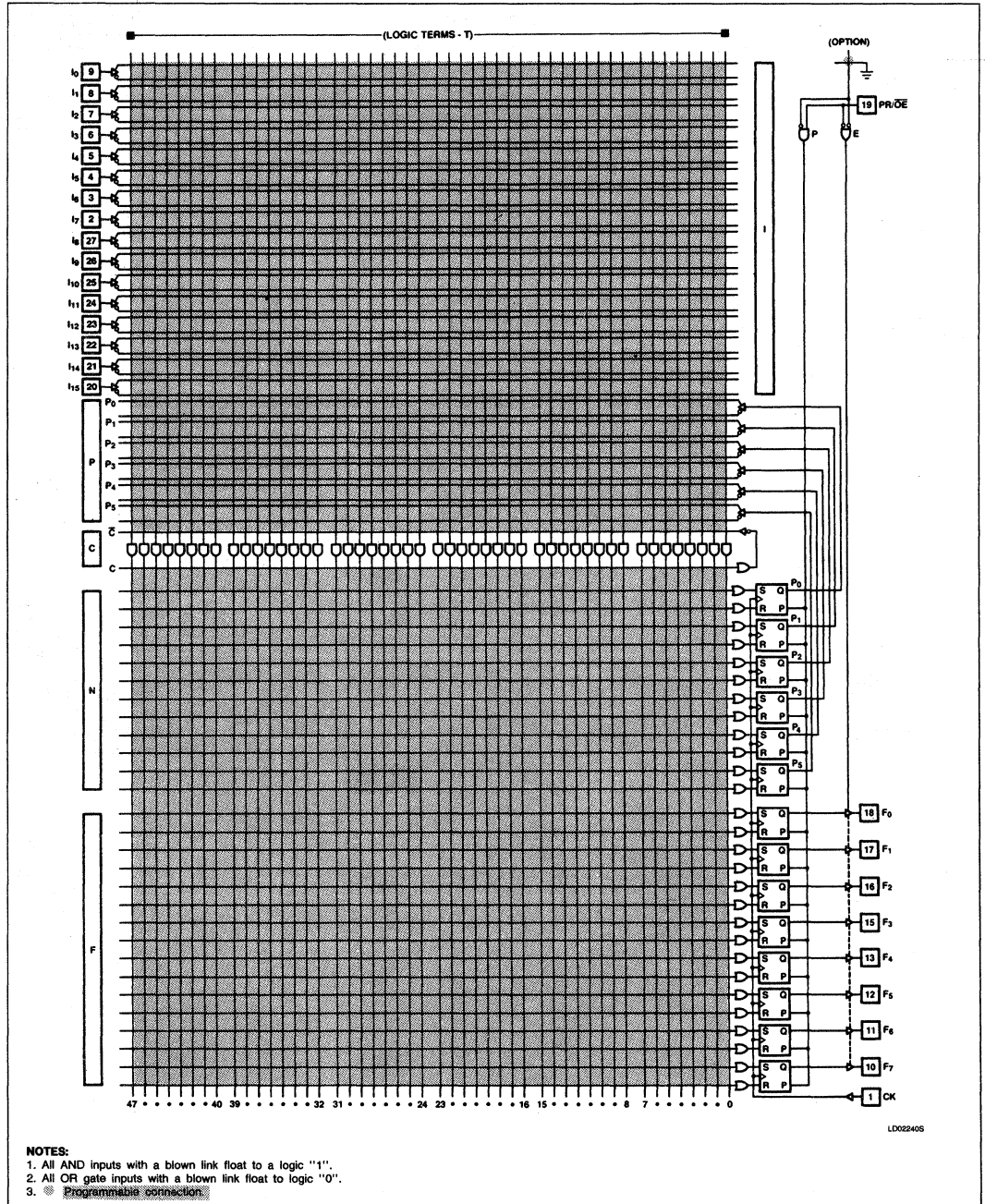
NOTE: The Test Array pattern MUST be deleted before incorporating a user program. This is accomplished automatically when using Signetics qualified programming equipment.

6

# Field-Programmable Logic Sequencer (16 × 48 × 8)

PLS105

## FPLS LOGIC DIAGRAM



## Field-Programmable Logic Sequencer (16 × 48 × 8)

PLS105

## ORDERING INFORMATION

DESCRIPTION	ORDER CODE
28-pin Plastic DIP 600mil-wide	PLS105N
28-pin Plastic Leaded Chip Carrier	PLS105A

## THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS<sup>1</sup>

SYMBOL	PARAMETER	RATING		UNIT
		Min	Max	
V <sub>CC</sub>	Supply voltage		+7	V <sub>DC</sub>
V <sub>IN</sub>	Input voltage		+5.5	V <sub>DC</sub>
V <sub>OUT</sub>	Output voltage		+5.5	V <sub>DC</sub>
I <sub>IN</sub>	Input currents	-30	+30	mA
I <sub>OUT</sub>	Output currents		+100	mA
T <sub>A</sub> T <sub>STG</sub>	Temperature range Operating Storage	0 -65	+75 +150	°C

DC ELECTRICAL CHARACTERISTICS 0°C ≤ T<sub>A</sub> ≤ 75°C, 4.75V ≤ V<sub>CC</sub> ≤ 5.25V

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ <sup>2</sup>	Max	
<b>Input voltage<sup>3</sup></b>						
V <sub>IH</sub> V <sub>IL</sub> V <sub>IC</sub>	High Low Clamp <sup>4</sup>	V <sub>CC</sub> = Max V <sub>CC</sub> = Min V <sub>CC</sub> = Min, I <sub>IN</sub> = -12mA	2	-0.8	0.8 -1.2	V
<b>Output voltage<sup>3</sup></b>						
V <sub>OH</sub> V <sub>OL</sub>	High <sup>5</sup> Low <sup>6</sup>	V <sub>CC</sub> = Min I <sub>OH</sub> = -2mA I <sub>OL</sub> = 9.6mA	2.4	0.35	0.45	V
<b>Input current</b>						
I <sub>IH</sub> I <sub>IL</sub> I <sub>IL</sub>	High Low Low (CK input)	V <sub>IN</sub> = 5.5V V <sub>IN</sub> = 0.45V V <sub>IN</sub> = 0.45V		< 1 -10 -50	25 -100 -250	μA
<b>Output current</b>						
I <sub>O(OFF)</sub> I <sub>OS</sub>	Hi-Z state <sup>7</sup> Short circuit <sup>4,8</sup>	V <sub>CC</sub> = Max V <sub>OUT</sub> = 5.5V V <sub>OUT</sub> = 0.45V V <sub>OUT</sub> = 0V		1 -1	40 -40 -70	μA mA
I <sub>CC</sub>	V <sub>CC</sub> supply current <sup>9</sup>	V <sub>CC</sub> = Max		120	180	mA
<b>Capacitance<sup>7</sup></b>						
C <sub>IN</sub> C <sub>OUT</sub>	Input Output	V <sub>CC</sub> = 5.0V V <sub>IN</sub> = 2.0V V <sub>OUT</sub> = 2.0V		8 10		pF

See Notes on following page.

6

## Field-Programmable Logic Sequencer (16 × 48 × 8)

PLS105

**NOTES:**

- Stresses above those listed under "Absolute Maximum Ratings" may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.
- All typical values are at  $V_{CC} = 5V$ ,  $T_A = +25^\circ C$ .
- All voltage values are with respect to network ground terminal.
- Test one at a time.
- Measured with  $V_{IL}$  applied to  $\overline{OE}$  and a logic high stored, or with  $V_{IH}$  applied to PR.
- Measured with a programmed logic condition for which the output is at a low logic level, and  $V_{IL}$  applied to PR/ $\overline{OE}$ . Output sink current is supplied through a resistor to  $V_{CC}$ .
- Measured with  $V_{IH}$  applied to PR/ $\overline{OE}$ .
- Duration of short circuit should not exceed 1 second.
- $I_{CC}$  is measured with the PR/ $\overline{OE}$  input grounded, all other inputs at 4.5V and the outputs open.

**AC ELECTRICAL CHARACTERISTICS**  $R_1 = 470\Omega$ ,  $R_2 = 1k\Omega$ ,  $C_L = 30pF$ ,  $0^\circ C \leq T_A \leq +75^\circ C$ ,  $4.75V \leq V_{CC} \leq 5.25V$ 

SYMBOL	PARAMETER	TO	FROM	LIMITS			UNIT
				Min	Typ <sup>1</sup>	Max	
<b>Pulse width</b>							
$t_{CKH}$	Clock <sup>2</sup> high	CK -	CK +	25	15		ns
$t_{CKL}$	Clock low	CK +	CK -	25	15		
$t_{CKP1B}$	Period (without Complement Array)	CK +	CK +	80	40		
$t_{CKP2B}$	Period (with Complement Array)	CK +	CK +	120	60		
$t_{PRH}$	Preset pulse	PR -	PR +	25	15		
<b>Setup time<sup>3</sup></b>							
$t_{S1A}$	Input	CK +	Input $\pm$	60			ns
$t_{S1B}$	Input	CK +	Input $\pm$	50			
$t_{S1C}$	Input	CK +	Input $\pm$	42			
$t_{S2A}$	Input (through Complement Array)	CK +	Input $\pm$	90			
$t_{S2B}$	Input (through Complement Array)	CK +	Input	80			
$t_{S2C}$	Input (through Complement Array)	CK +	Input	72			
$t_{VS}$	Power-on preset	CK -	$V_{CC} +$	0	-10		
$t_{PRS}$	Preset	CK -	PR -	0	-10		
<b>Hold time</b>							
$t_{IH}$	Input	Input $\pm$	CK +	5	-10		ns
<b>Propagation delay</b>							
$t_{CKO}$	Clock	Output $\pm$	CK +		15	30	ns
$t_{OE}$	Output enable	Output -	OE -		20	30	
$t_{OD}$	Output disable	Output +	OE +		20	30	
$t_{PR}$	Preset	Output +	PR +		18	30	
$t_{PPR}$	Power-on preset	Output +	$V_{CC} +$		0	10	
<b>Frequency of operation<sup>3</sup></b>							
$f_{MAXC}$	Without Complement Array					13.9	MHz
$f_{MAXC}$	With Complement Array					9.8	

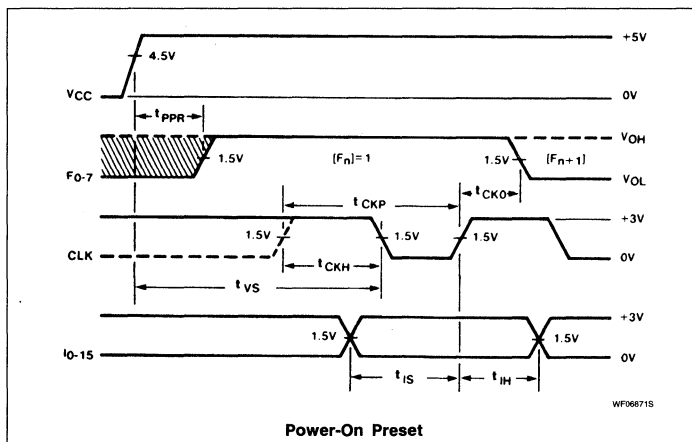
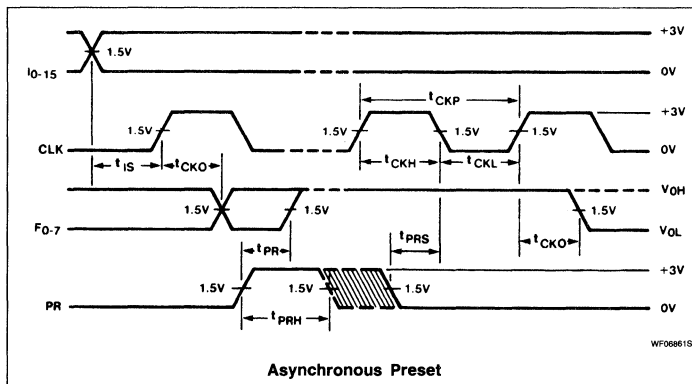
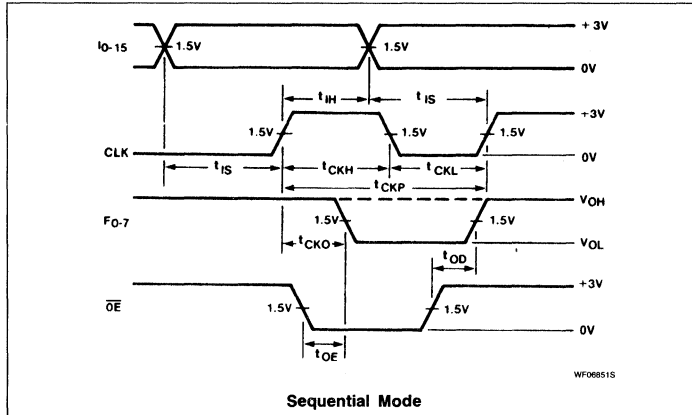
**NOTES:**

- All typical values are at  $V_{CC} = 5V$ ,  $T_A = +25^\circ C$ .
- To prevent spurious clocking, clock rise time (10% - 90%)  $\leq 30ns$ .
- See "Speed vs. OR Loading" diagrams.

# Field-Programmable Logic Sequencer (16 × 48 × 8)

PLS105

## TIMING DIAGRAMS



## TIMING DEFINITIONS

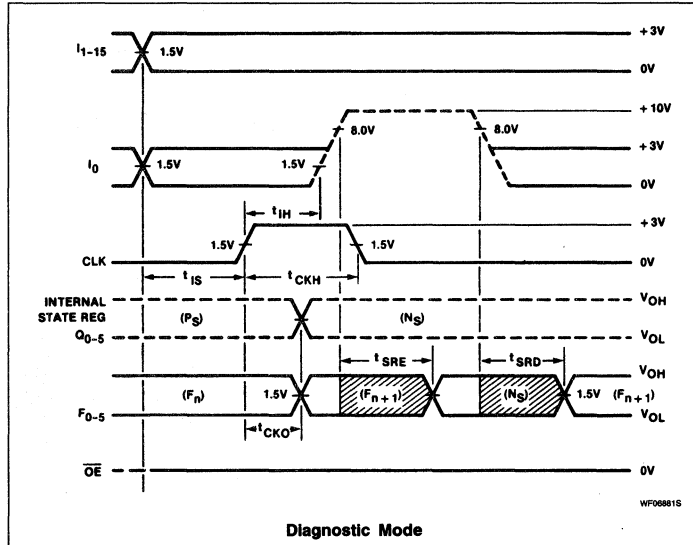
- $t_{CKH}$  Width of input clock pulse.
- $t_{CKL}$  Interval between clock pulses.
- $t_{CKP1}$  Clock period – when not using Complement Array.
- $t_{IS1}$  Required delay between beginning of valid input and positive transition of Clock.
- $t_{CKP2}$  Clock period – when using Complement Array.
- $t_{IS2}$  Required delay between beginning of valid input and positive transition of Clock, when using optional Complement Array (two passes necessary through the AND Array).
- $t_{VS}$  Required delay between  $V_{CC}$  (after power-on) and negative transition of Clock preceding first reliable clock pulse.
- $t_{PRS}$  Required delay between negative transition of Asynchronous Preset and negative transition of Clock preceding first reliable clock pulse.
- $t_{IH}$  Required delay between positive transition of Clock and end of valid Input data.
- $t_{CKO}$  Delay between positive transition of clock and when outputs become valid (with PR/OE low).
- $t_{OE}$  Delay between beginning of Output Enable Low and when Outputs become valid.
- $t_{OD}$  Delay between beginning of Output Enable High and when Outputs are in the off state.
- $t_{SRE}$  Delay between input  $I_0$  transition to Diagnostic Mode and when the Outputs reflect the contents of the State Register.
- $t_{SRD}$  Delay between input  $I_0$  transition to Logic mode and when the Outputs reflect the contents of the Output Register.
- $t_{PR}$  Delay between positive transition of Preset and when Outputs become valid at "1".
- $t_{PPR}$  Delay between  $V_{CC}$  (after power-on) and when Outputs become preset at "1".
- $t_{PRH}$  Width of preset input pulse.
- $f_{MAX}$  Maximum clock frequency.



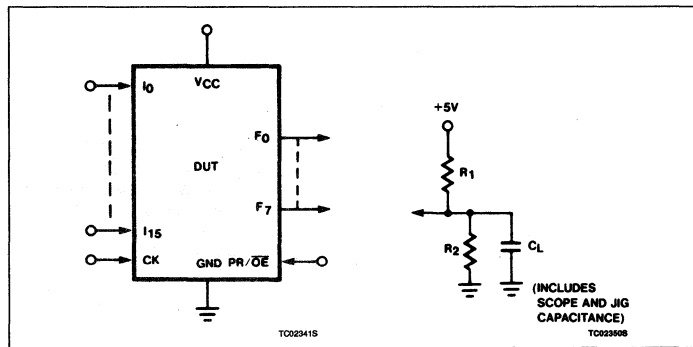
# Field-Programmable Logic Sequencer (16 × 48 × 8)

PLS105

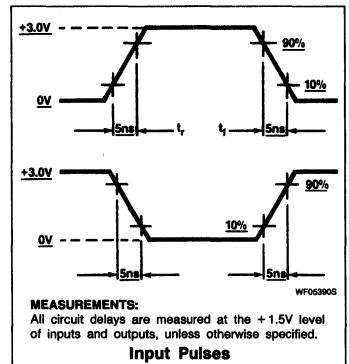
## TIMING DIAGRAMS (Continued)



## TEST LOAD CIRCUIT



## VOLTAGE WAVEFORM



# Field-Programmable Logic Sequencer (16 × 48 × 8)

PLS105

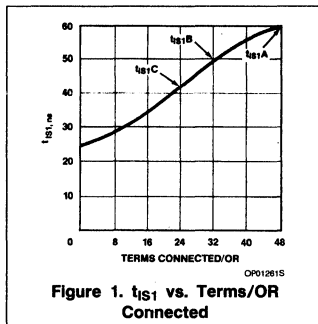
## SPEED VS. "OR" LOADING

The maximum frequency at which the FPLS can be clocked while operating in *sequential mode* is given by:

$$(1/f_{MAX}) = t_{CY} = t_{IS} + t_{CKO}$$

This frequency depends on the number of transition terms  $T_n$  used. Having all 48 terms connected in the AND array does not appreciably impact performance; but the number of terms connected to each OR line affects  $t_{IS}$ , due to capacitive loading. The effect of this loading can be seen in Figure 1, showing the variation of  $t_{IS1}$  with the number of terms connected per OR.

The AC electrical characteristics contain three limits for the parameters  $t_{IS1}$  and  $t_{IS2}$ . The first,  $t_{IS1A}$  is guaranteed for a device with 48 terms connected to any OR line.  $t_{IS1B}$  is guaranteed for a device with 32 terms connected to any OR line. And  $t_{IS1C}$  is guaranteed for a device with 24 terms connected to any OR line.



**Figure 1.  $t_{IS1}$  vs. Terms/OR Connected**

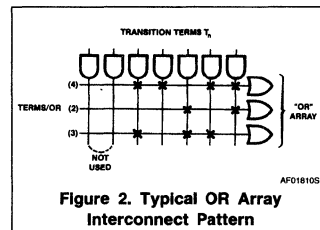
The three other entries in the AC table,  $t_{IS2}$  A, B, and C are corresponding 48, 32, and 24 term limits when using the on-chip Complement Array.

The worst case  $t_{IS}$  for a given application can be determined by identifying the OR line with the maximum number of  $t_n$  connections. This can be done by referring to

the interconnect pattern in the FPLS logic diagram, typically illustrated in Figure 2, or by counting the maximum number of "H" or "L" entries in one of the columns of the device Program Table.

This number plotted on the curve in Figure 1 will yield the worst case  $t_{IS}$  and, by implication, the maximum clocking frequency for reliable operation.

*Note that for maximum speed all UNUSED transition terms should be disconnected from the OR array.*



**Figure 2. Typical OR Array Interconnect Pattern**

# Field-Programmable Logic Sequencer (16 × 48 × 8)

# PLS105

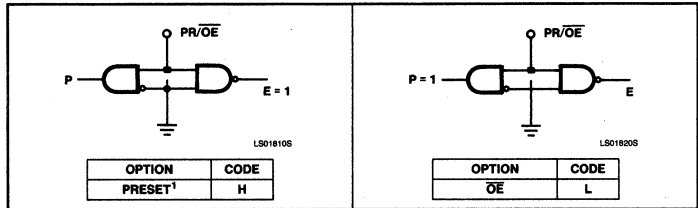
## LOGIC PROGRAMMING

The FPLS can be programmed by means of Logic programming equipment.

With Logic programming, the AND/OR gate input connections necessary to implement the desired logic function are coded directly from the State Diagram using the Program Table on the following page.

In this Table, the logic state or action of control variables C, I, P, N, and F, associated with each Transition Term  $T_n$ , is assigned a symbol which results in the proper fusing pattern of corresponding links, defined as follows:

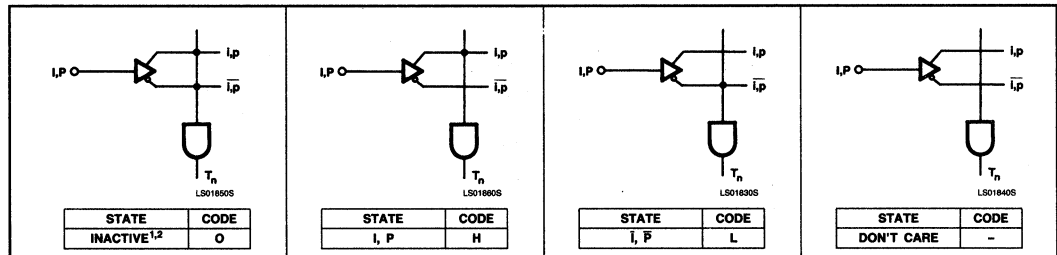
## PRESET/ $\overline{OE}$ OPTION - (P/E)



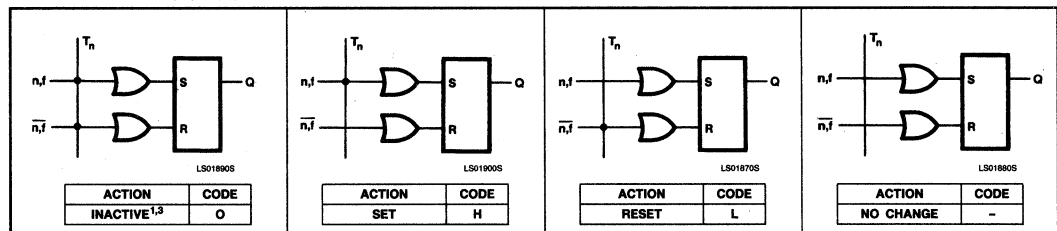
## PROGRAMMING THE PLS105:

The PLS105 has a power-up preset feature. This feature insures that the device will power-up in a known state with all register elements (state and output register) at a logic High (H). When programming the device it is important to realize this is the initial state of the device. You *must* provide a next state jump if you do not wish to use all Highs (H) as the present state.

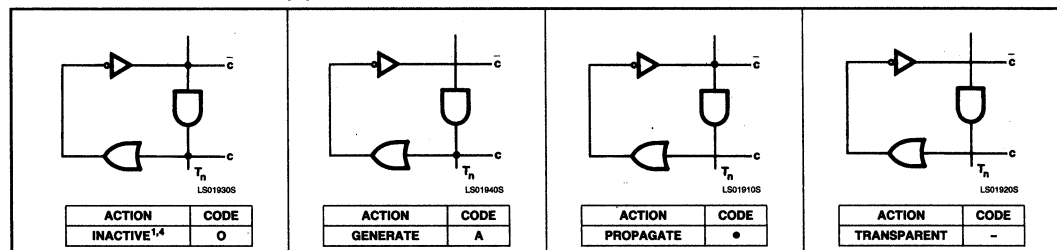
## "AND" ARRAY - (I), (P)



## "OR" ARRAY - (N), (F)



## "COMPLEMENT" ARRAY - (C)



### NOTES:

1. This is the initial unprogrammed state of all links.
2. Any gate  $T_n$  will be unconditionally inhibited if both the true and complement fuses of any input (I,P) are left intact.
3. To prevent simultaneous Set and Reset flip-flop commands, this state is not allowed for N and F link pairs coupled to active gates  $T_n$  (see flip-flop truth tables).
4. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates  $T_n$ .



# Field-Programmable Logic Sequencer (16 × 48 × 8)

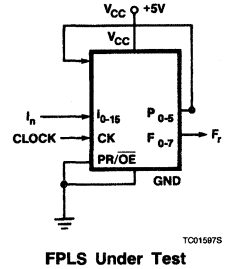
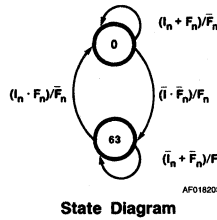
PLS105

## TEST ARRAY

The FPLS may be subjected to AC and DC parametric tests prior to programming via an on-chip test array.

The array consists of test transition terms 48 and 49, factory programmed as shown below.

Testing is accomplished by clocking the FPLS and applying the proper input sequence to  $I_{0-15}$  as shown in the test circuit timing diagram.



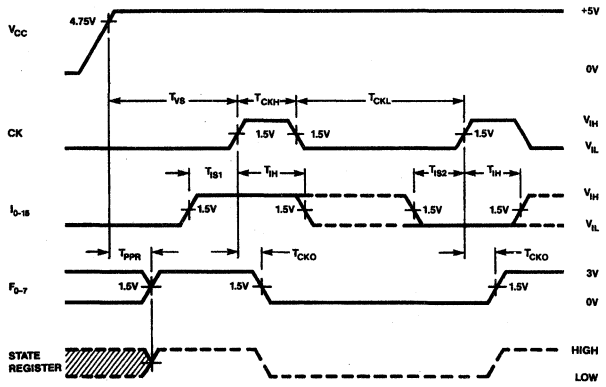
### TEST ARRAY PROGRAM

T E R M	C	AND																PRESENT STATE (Ps)															
		INPUT (Im)																															
		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	5	4	3	2	1	0	5	4	3	2	1	0				
48	A	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H						
49	●	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L						

OPTION (P/E)																H										
OR																										
NEXT STATE (Ns)								OUTPUT (Ff)																		
5	4	3	2	1	0	7	6	5	4	3	2	1	0	5	4	3	2	1	0							
L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	H
H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	

Test Array Program

Both terms 48 and 49 must be deleted during user programming to avoid interfering with the desired logic function. This is accomplished automatically by any of Signetics' qualified programming equipment.



Test Circuit Timing Diagram

### TEST ARRAY DELETED

T E R M	C	AND																PRESENT STATE (Ps)															
		INPUT (Im)																															
		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	5	4	3	2	1	0	5	4	3	2	1	0					
48	—	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H							
49	●	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L							

OPTION (P/E)																H										
OR																										
NEXT STATE (Ns)								OUTPUT (Ff)																		
5	4	3	2	1	0	7	6	5	4	3	2	1	0	5	4	3	2	1	0							
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Test Array Deleted

## PLS105A Field-Programmable Logic Sequencer (16 × 48 × 8)

Signetics Programmable Logic  
Product Specification

Application Specific Products  
• Series 28

### DESCRIPTION

The PLS105A is a bipolar programmable state machine of the Mealy type. It contains logic AND-OR gate arrays with user programmable connections which control the inputs of on-chip State and Output Registers. These consist respectively of 6  $Q_P$ , and 8  $Q_F$  edge-triggered, clocked S/R flip-flops, with an Asynchronous Preset option. All flip-flops are unconditionally preset to "1" during power turn on.

The AND array combines 16 external inputs  $I_0$ - $I_{15}$  with 6 internal inputs  $P_0$ - $P_5$ , which are fed back from the State Registers to form up to 48 Transition terms (AND terms). All Transition terms can include True, False, or Don't Care states of the controlling variables, and are merged in the OR array to issue next-state and next-output commands to their respective registers on the Low-to-High transition of the Clock pulse. Both True and Complement Transition terms can be generated by optional use of the internal input variable (C) from the Complement Array. Also, if desired, the Preset input can be converted to Output-Enable function, as an additional user programmable option.

Order codes are contained on the pages following.

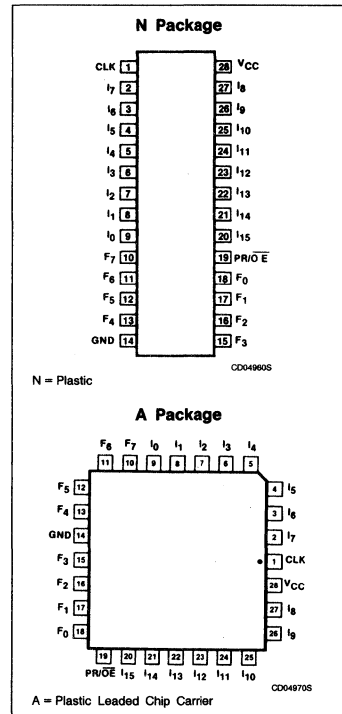
### FEATURES

- Field-Programmable (Ni-Cr link)
- 16 input variables
- 8 output functions
- 48 transition terms
- 6-bit State Register
- 8-bit Output Register
- Transition complement array
- Positive edge-triggered clocked flip-flops
- Programmable Asynchronous Preset or Output Enable
- Power-on preset to all "1" of internal registers
- $f_{MAX} = 20\text{MHz}$
- Power dissipation: 600mW (typ.)
- TTL compatible
- Single +5V supply
- Tri-state outputs

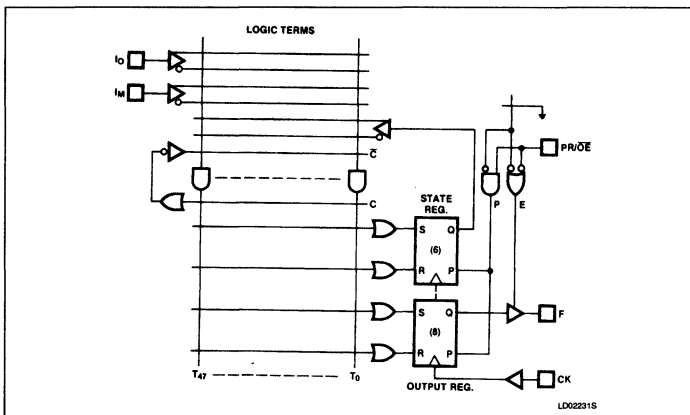
### APPLICATIONS

- Interface protocols
- Sequence detectors
- Peripheral controllers
- Timing generators
- Sequential circuits
- Elevator controllers
- Security locking systems
- Counters
- Shift registers

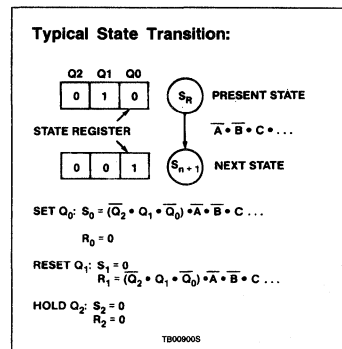
### PIN CONFIGURATIONS



### FUNCTIONAL DIAGRAM



### LOGIC FUNCTION



# Field-Programmable Logic Sequencer (16 × 48 × 8)

PLS105A

## PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION	POLARITY
1	CK	<b>Clock:</b> The Clock input to the State and Output Registers. A Low-to-High transition on this line is necessary to update the contents of both registers.	Active-High
2-8 20-27	I <sub>1-15</sub>	<b>Logic Inputs:</b> The 15 external inputs to the AND array used to program jump conditions between machine states, as determined by a given logic sequence.	Active-High/Low
9	I <sub>0</sub>	<b>Logic/Diagnostic Input:</b> A 16th external logic input to the AND array, as above, when exercised with standard TTL levels. When I <sub>0</sub> is held at +10V, device outputs F <sub>0-5</sub> reflect the contents of State Register bits P <sub>0-5</sub> . The contents of each Output Register remains unaltered.	Active-High/Low
10-13 15-18	F <sub>0-7</sub>	<b>Logic/Diagnostic Outputs:</b> Eight device outputs which normally reflect the contents of Output Register bits Q <sub>0-7</sub> , when enabled. When I <sub>0</sub> is held at +10V, F <sub>0-5</sub> = (P <sub>0-5</sub> ), and F <sub>6,7</sub> = Logic "1".	Active-High
19	PR/ $\overline{OE}$	<b>Preset or Output Enable Input:</b> A user programmable function: <ul style="list-style-type: none"> <li><b>Preset:</b> Provides an Asynchronous Preset to logic "1" of all State and Output Register bits. Preset overrides Clock, and when held High, clocking is inhibited and F<sub>0-7</sub> are High. Normal clocking resumes with the first full clock pulse following a High-to-Low clock transition, after Preset goes Low.</li> <li><b>Output Enable:</b> Provides an output enable function to buffers F<sub>0-7</sub> from the Output Register.</li> </ul>	Active-High (H)  Active-Low (L)

## TRUTH TABLE 1, 2, 3, 4, 5, 6

V <sub>cc</sub>	OPTION		I <sub>0</sub>	CK	S	R	Q <sub>P/F</sub>	F
	PR	$\overline{OE}$						
+5V	H		*	X	X	X	H	H
	L		+10V	X	X	X	Q <sub>n</sub>	(Q <sub>P</sub> ) <sub>n</sub>
	L		X	X	X	X	Q <sub>n</sub>	(Q <sub>F</sub> ) <sub>n</sub>
		H	*	X	X	X	Q <sub>n</sub>	Hi-Z
		L	+10V	X	X	X	Q <sub>n</sub>	(Q <sub>P</sub> ) <sub>n</sub>
		L	X	X	X	X	Q <sub>n</sub>	(Q <sub>F</sub> ) <sub>n</sub>
		L	X	↑	L	L	Q <sub>n</sub>	(Q <sub>F</sub> ) <sub>n</sub>
		L	X	↑	L	H	L	L
		L	X	↑	H	L	H	H
		L	X	↑	H	H	IND.	IND.
↑	X	X	X	X	X	X	H	

### NOTES:

- Positive Logic:  
 $S/R = T_0 + T_1 + T_2 + \dots + T_{47}$   
 $T_n = C(I_0 I_1 I_2 \dots) (P_0 P_1 \dots P_5)$
- Either Preset (Active-High) or Output Enable (Active-Low) are available, but not both. The desired function is a user-programmable option.
- ↑ denotes transition from Low to High level.
- R = S = High is an illegal input condition.
- \* = H/L/+10V
- X = Don't Care ( $\leq 5.5V$ )

## VIRGIN STATE

A factory shipped virgin device contains all fusible links intact, such that:

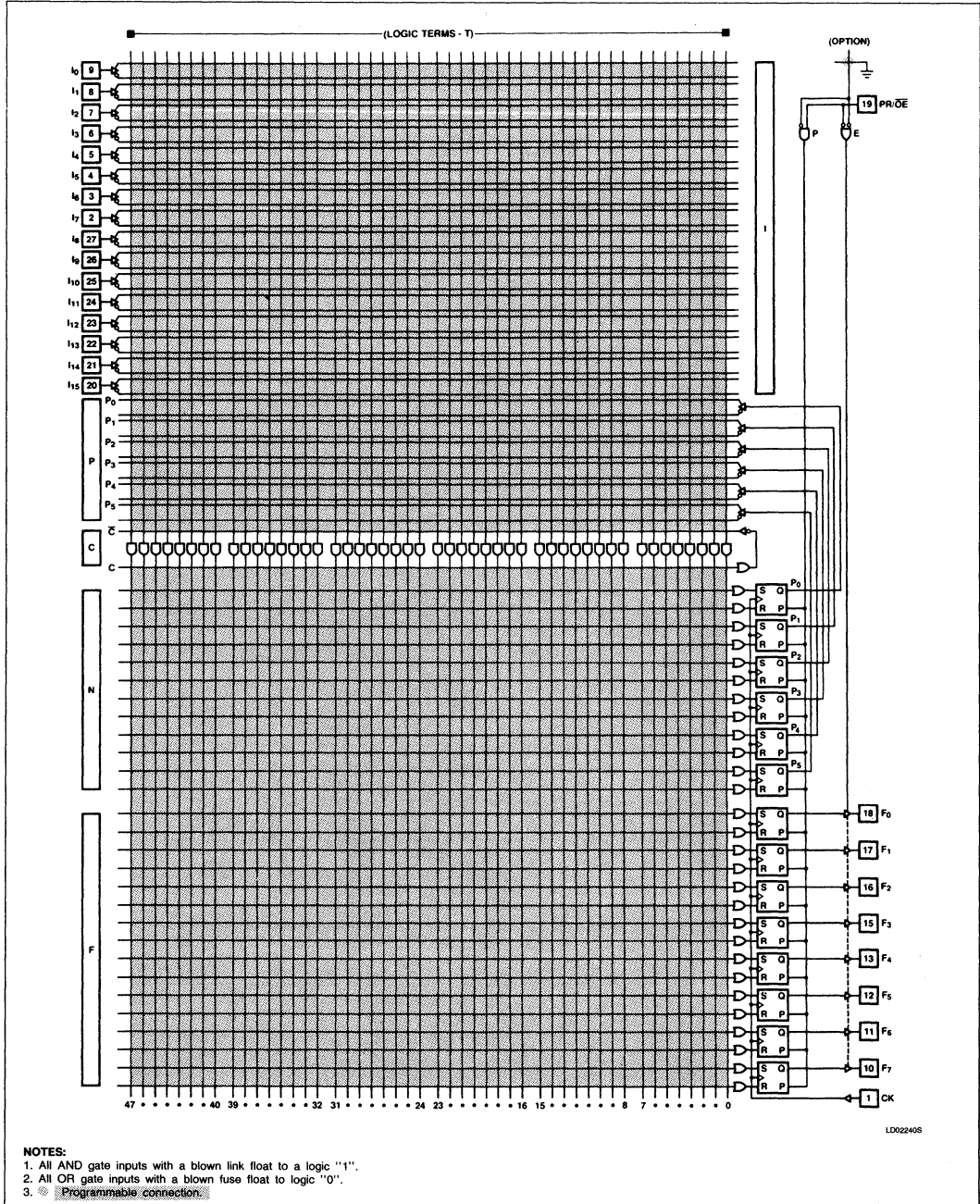
- PR/ $\overline{OE}$  option is set to PR. Thus, all outputs will be at "1", as preset by initial power-up procedure.
- All transition terms are disabled (0).
- All S/R flip-flop inputs are disabled (0).
- The device can be clocked via a Test Array pre-programmed with a standard test pattern.

NOTE: The Test Array pattern MUST be deleted before incorporating a user program. This is accomplished automatically when using any of Signetics' qualified programming equipment.

# Field-Programmable Logic Sequencer (16 × 48 × 8)

PLS105A

## FPLS LOGIC DIAGRAM



6



## Field-Programmable Logic Sequencer (16 × 48 × 8)

PLS105A

## ORDERING INFORMATION

DESCRIPTION	ORDER CODE
28-pin Plastic DIP 600mil-wide	PLS105AN
28-pin Plastic Leaded Chip Carrier	PLS105AA

## THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS<sup>1</sup>

SYMBOL	PARAMETER	RATING		UNIT
		Min	Max	
V <sub>CC</sub>	Supply voltage		+7	V <sub>DC</sub>
V <sub>IN</sub>	Input voltage		+5.5	V <sub>DC</sub>
V <sub>OUT</sub>	Output voltage		+5.5	V <sub>DC</sub>
I <sub>IN</sub>	Input currents	-30	+30	mA
I <sub>OUT</sub>	Output currents		+100	mA
T <sub>A</sub> T <sub>STG</sub>	Temperature range Operating Storage	0 -65	+75 +150	°C

DC ELECTRICAL CHARACTERISTICS 0°C ≤ T<sub>A</sub> ≤ 75°C, 4.75V ≤ V<sub>CC</sub> ≤ 5.25V

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ <sup>2</sup>	Max	
<b>Input voltage<sup>3</sup></b>						
V <sub>IH</sub> V <sub>IL</sub> V <sub>IC</sub>	High Low Clamp <sup>4</sup>	V <sub>CC</sub> = Max V <sub>CC</sub> = Min V <sub>CC</sub> = Min, I <sub>IN</sub> = -12mA	2		0.8 -1.2	V
<b>Output voltage<sup>3</sup></b>						
V <sub>OH</sub> V <sub>OL</sub>	High <sup>5</sup> Low <sup>6</sup>	V <sub>CC</sub> = Min I <sub>OH</sub> = -2mA I <sub>OL</sub> = 9.6mA	2.4	0.35	0.45	V
<b>Input current</b>						
I <sub>IH</sub> I <sub>IL</sub> I <sub>IL</sub>	High Low Low (CK input)	V <sub>IN</sub> = 5.5V V <sub>IN</sub> = 0.45V V <sub>IN</sub> = 0.45V		< 1 -10 -50	25 -100 -250	μA
<b>Output current</b>						
I <sub>O(OFF)</sub> I <sub>OS</sub>	Hi-Z state <sup>7</sup> Short circuit <sup>4,8</sup>	V <sub>CC</sub> = Max V <sub>OUT</sub> = 5.5V V <sub>OUT</sub> = 0.45V V <sub>OUT</sub> = 0V		1 -1	40 -40 -70	μA mA
I <sub>CC</sub>	V <sub>CC</sub> supply current <sup>9</sup>	V <sub>CC</sub> = Max		120	180	mA
<b>Capacitance<sup>7</sup></b>						
C <sub>IN</sub> C <sub>OUT</sub>	Input Output	V <sub>CC</sub> = 5.0V V <sub>IN</sub> = 2.0V V <sub>OUT</sub> = 2.0V		8 10		pF

See Notes on following page.

## Field-Programmable Logic Sequencer (16 × 48 × 8)

PLS105A

**NOTES:**

- Stresses above those listed under "Absolute Maximum Ratings" may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.
- All typical values are at  $V_{CC} = 5V$ ,  $T_A = +25^\circ C$ .
- All voltage values are with respect to network ground terminal.
- Test one at a time.
- Measured with  $V_{IL}$  applied to  $\overline{OE}$  and a logic high stored, or with  $V_{IH}$  applied to PR.
- Measured with a programmed logic condition for which the output is at a low logic level, and  $V_{IL}$  applied to PR/ $\overline{OE}$ . Output sink current is supplied through a resistor to  $V_{CC}$ .
- Measured with  $V_{IH}$  applied to PR/ $\overline{OE}$ .
- Duration of short circuit should not exceed 1 second.
- $I_{CC}$  is measured with the PR/ $\overline{OE}$  input grounded, all other inputs at 4.5V and the outputs open.

**AC ELECTRICAL CHARACTERISTICS**  $R_1 = 470\Omega$ ,  $R_2 = 1k\Omega$ ,  $C_L = 30pF$ ,  $0^\circ C \leq T_A \leq +75^\circ C$ ,  $4.75V \leq V_{CC} \leq 5.25V$ 

SYMBOL	PARAMETERS	TO	FROM	LIMITS			UNIT
				Min	Typ <sup>1</sup>	Max	
<b>Pulse width</b>							
$t_{CKH}$	Clock <sup>2</sup> high	CK -	CK +	25	15		ns
$t_{CKL}$	Clock low	CK +	CK -	25	15		
$t_{CKP1B}$	Period (without Complement Array)	CK +	CK +	50	40		
$t_{CKP2B}$	Period (with Complement Array)	CK +	CK +	80	50		
$t_{PRH}$	Preset pulse	PR -	PR +	25	15		
<b>Setup time<sup>3</sup></b>							
$t_{IS1A}$	Input	CK +	Input $\pm$	40			ns
$t_{IS1B}$	Input	CK +	Input $\pm$	30			
$t_{IS2A}$	Input (through Complement Array)	CK +	Input $\pm$	70			
$t_{IS2B}$	Input (through Complement Array)	CK +	Input	60			
$t_{VS}$	Power-on preset	CK -	$V_{CC} +$	0	-10		
$t_{PRS}$	Preset	CK -	PR -	0	-10		
<b>Hold time</b>							
$t_{IH}$	Input	Input $\pm$	CK +	5	-10		ns
<b>Propagation delay</b>							
$t_{CKO}$	Clock	Output $\pm$	CK +		15	20	ns
$t_{OE}$	Output enable	Output -	OE -		20	30	
$t_{OD}$	Output disable	Output +	OE +		20	30	
$t_{PR}$	Preset	Output +	PR +		18	30	
$t_{PPR}$	Power-on preset	Output +	$V_{CC} +$		0	10	
<b>Frequency of operation</b>							
$f_{MAXB}$	Without Complement Array					20	MHz
$f_{MAXB}$	With Complement Array					12.5	

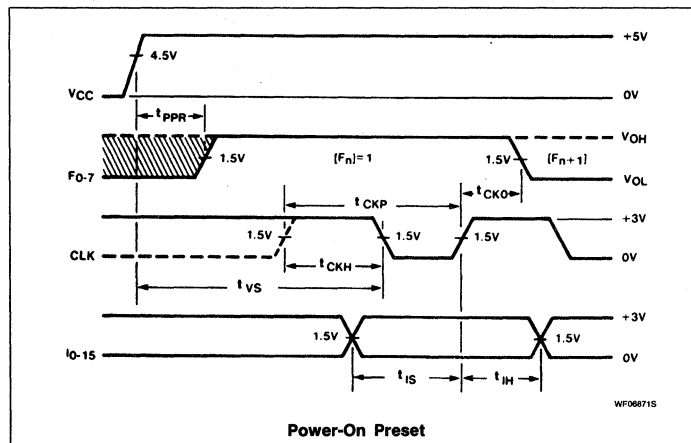
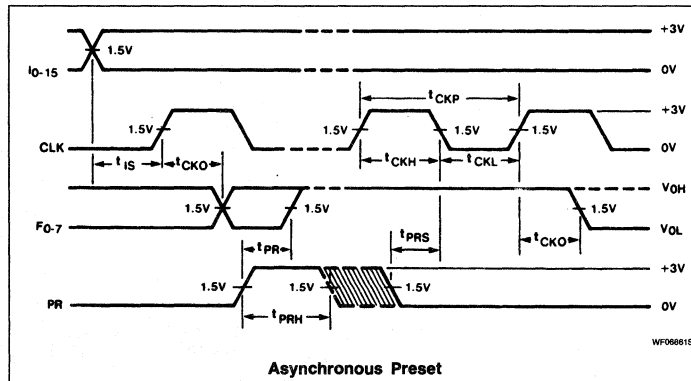
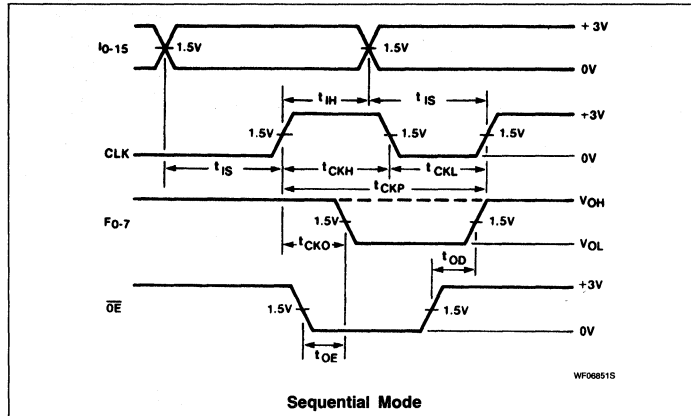
**NOTES:**

- All typical values are at  $V_{CC} = 5V$ ,  $T_A = +25^\circ C$ .
- To prevent spurious clocking, clock rise time (10% - 90%)  $\leq 30ns$ .
- See "Speed vs. OR Loading" diagrams.

# Field-Programmable Logic Sequencer (16 × 48 × 8)

# PLS105A

## TIMING DIAGRAMS



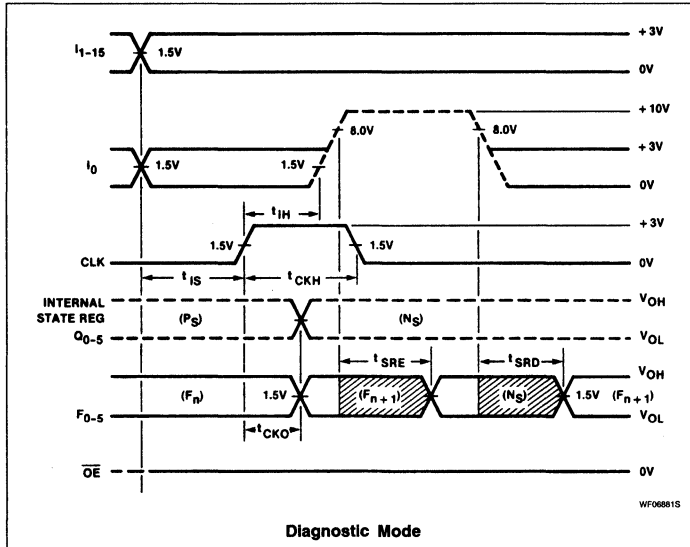
## TIMING DEFINITIONS

- $t_{CKH}$  Width of input clock pulse.
- $t_{CKL}$  Interval between clock pulses.
- $t_{CKP}$  Clock period — when not using Complement Array.
- $t_{IS1}$  Required delay between beginning of valid input and positive transition of Clock.
- $t_{CKP2}$  Clock period — when using Complement Array.
- $t_{IS2}$  Required delay between beginning of valid input and positive transition of Clock, when using optional Complement Array (two passes necessary through the AND Array).
- $t_{VS}$  Required delay between  $V_{CC}$  (after power-on) and negative transition of Clock preceding first reliable clock pulse.
- $t_{PRS}$  Required delay between negative transition of Asynchronous Preset and negative transition of Clock preceding first reliable clock pulse.
- $t_{IH}$  Required delay between positive transition of Clock and end of valid Input data.
- $t_{CKO}$  Delay between positive transition of Clock and when Outputs become valid (with PR/OE low).
- $t_{OE}$  Delay between beginning of Output Enable Low and when Outputs become valid.
- $t_{OD}$  Delay between beginning of Output Enable High and when Outputs are in the off state.
- $t_{SRE}$  Delay between input  $I_0$  transition to Diagnostic Mode and when the Outputs reflect the contents of the State Register.
- $t_{SRD}$  Delay between input  $I_0$  transition to Logic mode and when the Outputs reflect the contents of the Output Register.
- $t_{PR}$  Delay between positive transition of Preset and when Outputs become valid at "1".
- $t_{PPR}$  Delay between  $V_{CC}$  (after power-on) and when Outputs become preset at "1".
- $t_{PRH}$  Width of preset input pulse.
- $f_{MAX}$  Maximum clock frequency.

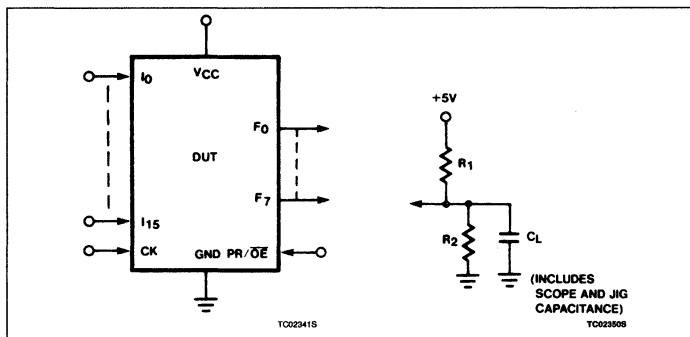
# Field-Programmable Logic Sequencer (16 × 48 × 8)

PLS105A

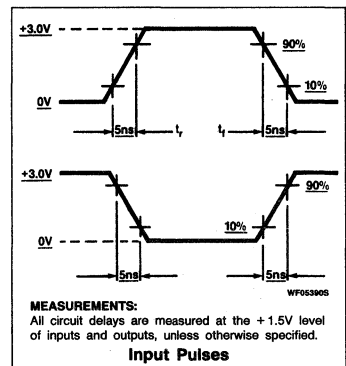
## TIMING DIAGRAMS (Continued)



## TEST LOAD CIRCUIT



## VOLTAGE WAVEFORM



# Field-Programmable Logic Sequencer (16 × 48 × 8)

PLS105A

## SPEED VS. "OR" LOADING

The maximum frequency at which the FPLS can be clocked while operating in *sequential mode* is given by:

$$(1/f_{MAX}) = t_{CY} = t_{IS} + t_{CKO}$$

This frequency depends on the number of transition terms  $T_n$  used. Having all 48 terms connected in the AND array does not appreciably impact performance; but the number of terms connected to each OR line affects  $t_{IS}$ , due to capacitive loading. The effect of this loading can be seen in Figure 1, showing the variation of  $t_{IS1}$  with the number of terms connected per OR.

The AC electrical characteristics contain two limits for the parameters  $t_{IS1}$  and  $t_{IS2}$ . The first,  $t_{IS1A}$  is guaranteed for a device with 24 terms connected to any OR line.  $t_{IS1B}$  is guaranteed for a device with 16 terms connected to any OR line.

The two other entries in the AC table,  $t_{IS2A}$  and B are corresponding 24 and 16 term limits when using the on-chip Complement array.

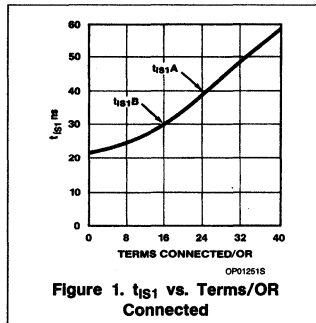


Figure 1.  $t_{IS1}$  vs. Terms/OR Connected

The worst case of  $t_{IS}$  for a given application can be determined by identifying the OR line with the maximum number of  $T_n$  connections. This can be done by referring to

the interconnect pattern in the FPLS logic diagram, typically illustrated in Figure 2, or by counting the maximum number of "H" or "L" entries in one of the columns of the device Program Table.

This number plotted on the curve in Figure 1 will yield the worst case  $t_{IS}$  and, by implication, the maximum clocking frequency for reliable operation.

Note that for maximum speed all UNUSED transition terms should be disconnected from the OR array.

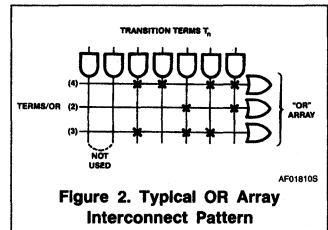


Figure 2. Typical OR Array Interconnect Pattern

# Field-Programmable Logic Sequencer (16 × 48 × 8)

PLS105A

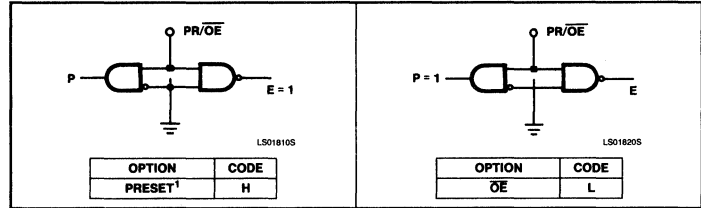
## LOGIC PROGRAMMING

The PLS can be programmed by means of Logic programming equipment.

With Logic programming, the AND/OR gate input connections necessary to implement the desired logic function are coded directly from the State Diagram using the Program Table of the following page.

In this Table, the logic state or action of control variables C, I, P, N, and F, associated with each Transition Term  $T_n$ , is assigned a symbol which results in the proper fusing pattern of corresponding links, defined as follows:

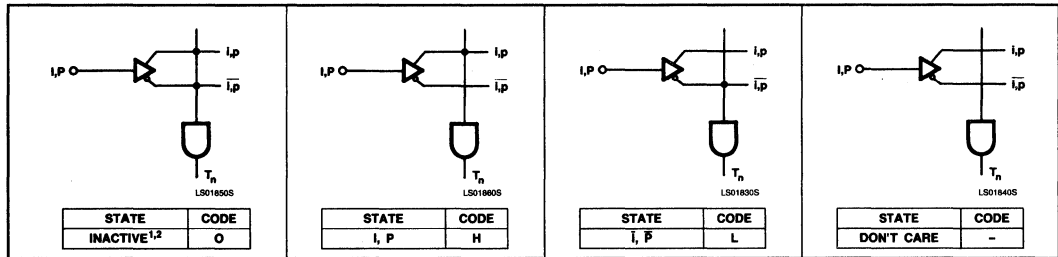
## PRESET/OE OPTION - (P/E)



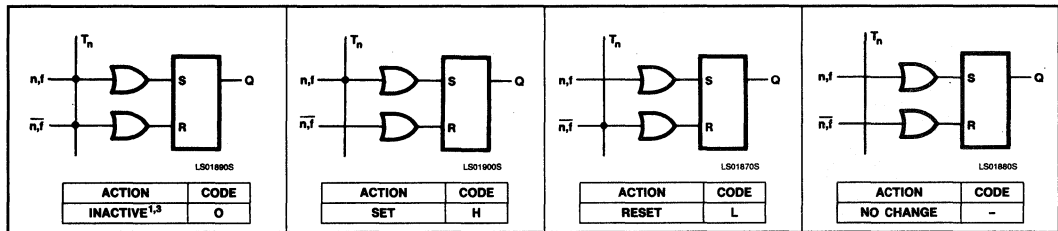
### PROGRAMMING THE PLS105A:

The PLS105A has a power-up preset feature. This feature insures that the device will power-up in a known state with all register elements (state and output register) at a logic High (H). When programming the device it is important to realize this is the initial state of the device. You *must* provide a next state jump if you do not wish to use all Highs (H) as the present state.

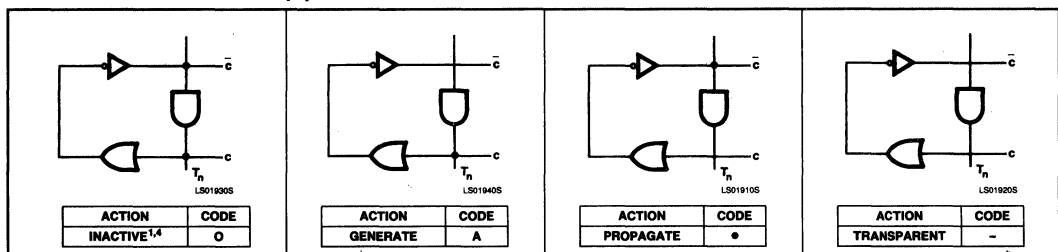
## "AND" ARRAY - (I), (P)



## "OR" ARRAY - (N), (F)



## "COMPLEMENT" ARRAY - (C)



### NOTES:

1. This is the initial unprogrammed state of all link pairs. It is normally associated with all unused (inactive) AND gates  $T_n$ .
2. Any gate  $T_n$  will be unconditionally inhibited if both the true and complement fuses of any input (I,P) are left intact.
3. To prevent simultaneous Set and Reset flip-flop commands, this state is not allowed for N and F link pairs coupled to active gates  $T_n$  (see flip-flop truth tables).
4. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates  $T_n$ .



# Field-Programmable Logic Sequencer (16 × 48 × 8)

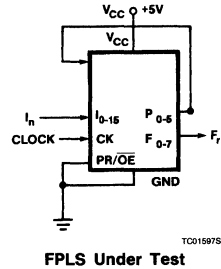
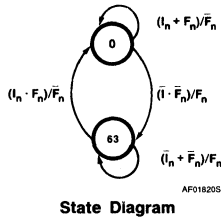
# PLS105A

### TEST ARRAY

The FPLS may be subjected to AC and DC parametric tests prior to programming via an on-chip test array.

The array consists of test transition terms 48 and 49, factory programmed as shown below.

Testing is accomplished by clocking the FPLS and applying the proper input sequence to  $I_{0-15}$  as shown in the test circuit timing diagram.



### TEST ARRAY PROGRAM

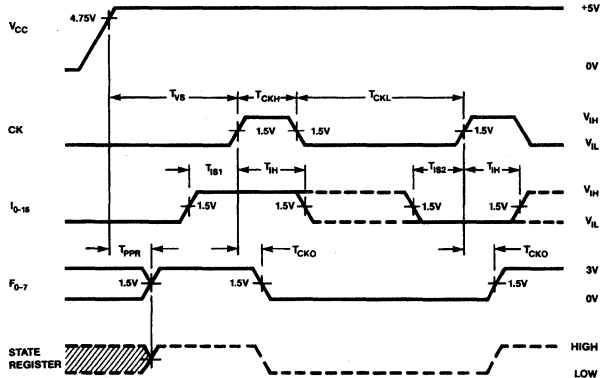
T E R M	C	AND																									
		INPUT ( $I_m$ )										PRESENT STATE ( $P_n$ )															
		1	1	1	1	1	1	0	9	8	7	6	5	4	3	2	1	0	5	4	3	2	1	0			
48	A	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
49	●	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L

OPTION (P/E)		OR		M										
NEXT STATE ( $N_n$ )		OUTPUT ( $F_f$ )												
5	4	3	2	1	0	7	6	5	4	3	2	1	0	
L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
H	H	H	H	H	H	H	H	H	H	H	H	H	H	H

Test Array Program

TB017805

Both terms 48 and 49 must be deleted during user programming to avoid interfering with the desired logic function. This is accomplished automatically by any of Signetics' qualified programming equipment.



### TEST ARRAY DELETED

T E R M	C	AND																									
		INPUT ( $I_m$ )										PRESENT STATE ( $P_n$ )															
		1	1	1	1	1	1	0	9	8	7	6	5	4	3	2	1	0	5	4	3	2	1	0			
48	—	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
49	●	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L

OPTION (P/E)		OR		M										
NEXT STATE ( $N_n$ )		OUTPUT ( $F_f$ )												
5	4	3	2	1	0	7	6	5	4	3	2	1	0	
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Test Array Deleted

TB017505

6



# PLUS405A

## Field-Programmable Logic Sequencer (16 × 64 × 8)

Signetics Programmable Logic  
Preliminary Specification

Application Specific Products  
• Series 28

### DESCRIPTION

The PLUS405A is a bipolar, programmable state machine of the Mealy type. Both the AND and the OR array are user programmable. All 64 AND gates are connected to the 16 external dedicated inputs (I<sub>0</sub> - I<sub>15</sub>) and to the feedback paths of the 8 on-chip State Registers (Q<sub>P0</sub> - Q<sub>P7</sub>). Both true and complement transition terms can be generated via optional use of the two internal Complement Arrays (input variables C<sub>0</sub>, C<sub>1</sub>).

All transition terms can include True, False and Don't Care states of the controlling state variables. All AND gates are merged into the programmable OR array to issue the next-state and next-output commands to their respective registers. Because the OR array is programmable, any one or all of the 64 transition terms can be connected to any or all of the State and Output Registers.

All state (Q<sub>P0</sub> - Q<sub>P7</sub>) and output (Q<sub>F0</sub> - Q<sub>F7</sub>) registers are edge-triggered, clocked J-K flip-flops, with Asynchronous Preset and Reset options. The PLUS405A architecture provides the added flexibility of the J-K toggle function which is indeterminate on S-R flip-flops. Each register may be individually programmed such that a specific Preset-Reset pattern is initialized when the initialization pin is raised to a logic level "1". (Note: Upon power-up, all registers are unconditionally preset to "1"). If desired, the initialization input pin (INIT) can be converted to an Output Enable (OE) function as an additional user programmable feature.

Availability of two user programmable clocks allows the user to design an independent counter or state machine function that can be used to control the bank of output registers.

Order codes are contained on the pages following.

### FEATURES

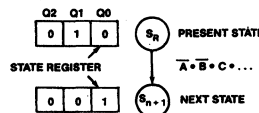
- Field-Programmable (Ti-W fusible link)
- 16 input variables
- 8 output functions
- 64 transition terms
- 8-bit State Register
- 8-bit Output Register
- 2 Transition complement array term
- Multiple clocks
- Programmable Asynchronous Preset/Reset or Output Enable
- Power-on preset to "1" of registers
- f<sub>MAX</sub> = 40MHz
- 950mW power dissipation (typ.)
- TTL compatible
- J-K or S-R flip-flop functions
- Tri-state outputs
- Functional superset of PLS105/105A

### APPLICATIONS

- Interface protocols
- Sequence detectors
- Peripheral controllers
- Timing generators
- Sequential circuits
- Elevator controllers
- Security locking systems
- Counters
- Shift registers

### LOGIC FUNCTION

#### Typical State Transition:



$$\text{SET } Q_i: S_i = (\bar{Q}_2 + Q_1 + \bar{Q}_0) \cdot \bar{A} \cdot \bar{B} + C \dots$$

$$R_i = 0$$

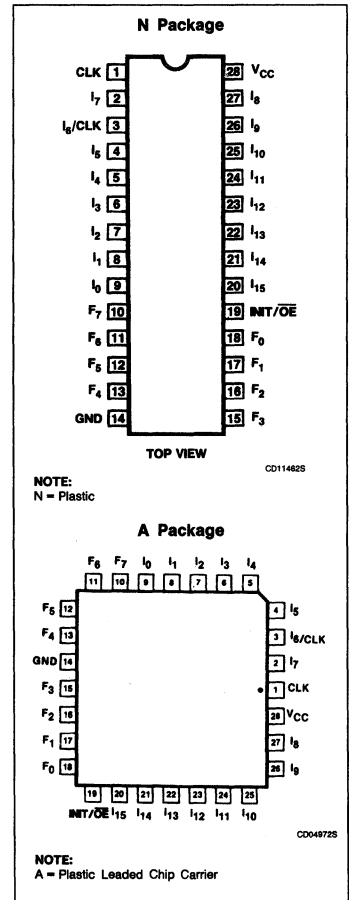
$$\text{RESET } Q_i: S_i = 0$$

$$R_i = (\bar{Q}_2 + Q_1 + \bar{Q}_0) \cdot \bar{A} \cdot \bar{B} + C \dots$$

$$\text{HOLD } Q_i: S_i = 0$$

$$R_i = 0$$

### PIN CONFIGURATIONS



NOTE:  
N = Plastic

CD114825

NOTE:  
A = Plastic Leaded Chip Carrier

CD048725

# Field-Programmable Logic Sequencer (16 × 64 × 8)

PLUS405A

## PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION	POLARITY
1	CK	<b>Clock:</b> The Clock input to the State and Output Registers. A Low-to-High transition on this line is necessary to update the contents of both registers.	Active – High (H)
2, 4–9, 26–27 21–22	$I_0–I_5, I_7$ $I_8–I_9$ $I_{13}–I_{14}$	<b>Logic Inputs:</b> The 11 external inputs to the AND array used to program jump conditions between machine states, as determined by a given logic sequence.	Active – High/Low (H/L)
3	$I_6/CK$	<b>Logic Input/Clock:</b> A user programmable function: • <b>Logic Input:</b> A 12th external logic input to the AND array, as above.  • <b>Clock:</b> A 2nd clock for the State Registers, as above. The Output Registers continue to be clocked off of Pin 1. If two separate State Machines are designed, setup and hold times must be adhered to in order to synchronize both banks of registers. Note that input buffer $I_6$ must be deleted from the AND array (i.e. all fuse locations "Don't Care") when using Pin 3 as a Clock.	Active – High/Low (H/L)  Active – High (H)
20	$I_{15}$	<b>High Fan-Out Logic Input:</b> A 13th external logic input to the AND array, as above. $I_{15}$ is capable of pulling down all 64 Product Terms by itself.	Active – High/Low (H/L)
23	$I_{12}$	<b>Logic/Diagnostic Input:</b> A 14th external logic input to the AND array, as above, when exercising standard TTL levels. When $I_{12}$ is held at +10V, device outputs $F_0–F_7$ reflect the contents of State Register bits $P_0–P_7$ . The contents of each Output Register remains unaltered.	Active – High/Low (H/L)
24	$I_{11}$	<b>Logic/Diagnostic Input:</b> A 15th external logic input to the AND array, as above, when exercising standard TTL levels. When $I_{11}$ is held at +10V, device outputs $F_0–F_7$ become direct inputs for State Register bits $P_0–P_7$ ; a Low-to-High transition on the appropriate clock line loads the values on pins $F_0–F_7$ into the State Register bits $P_0–P_7$ . The contents of each Output Register remains unaltered.	Active – High/Low (H/L)
25	$I_{10}$	<b>Logic/Diagnostic Input:</b> A 16th external logic input to the AND array, as above, when exercising standard TTL levels. When $I_{10}$ is held at +10V, device outputs $F_0–F_7$ become direct inputs for Output Register bits $Q_0–Q_7$ ; a Low-to-High transition on the appropriate clock line loads the values on pins $F_0–F_7$ into the Output Register bits $Q_0–Q_7$ . The contents of each State Register remains unaltered.	Active – High/Low (H/L)
10–13 15–18	$F_0–F_7$	<b>Logic Outputs/Diagnostic Outputs/Diagnostic Inputs:</b> Eight device outputs which normally reflect the contents of Output Register Bits $Q_0–Q_7$ , when enabled. When $I_{12}$ is held at +10V, $F_0–F_7 = (P_0–P_7)$ . When $I_{11}$ is held at +10V, $F_0–F_7$ become inputs to State Register bits $P_0–P_7$ . When $I_{10}$ is held at +10V, $F_0–F_7$ become inputs to Output Register bits $Q_0–Q_7$ .	Active – High (H)
19	INIT/ $\overline{OE}$	<b>Initialization or Output Enable Input:</b> <b>A user programmable function:</b> • <b>Initialization:</b> Provides an asynchronous preset to logic "1" or reset to logic "0" of all State and Output Register bits, determined individually for each register bit through user programming. INIT overrides Clock, and when held HIGH, clocking is inhibited and $F_0–F_7$ are in their initialization state. Normal clocking resumes with the first full clock pulse following a High-to-Low clock transition, after INIT goes low. • <b>Output Enable:</b> Provides an output enable function to buffers $F_0–F_7$ from the Output Registers.	Active – High (H)  Active – Low (L)

# Field-Programmable Logic Sequencer (16 × 64 × 8)

PLUS405A

**TRUTH TABLE 1, 2, 3, 4, 5, 6, 7**

V <sub>cc</sub>	OPTION		I <sub>10</sub>	I <sub>11</sub>	I <sub>12</sub>	CK	J	K	Q <sub>P</sub>	Q <sub>F</sub>	F
	INIT	$\overline{OE}$									
+5V	H	■	*	*	*	X	X	X	H/L	H/L	Q <sub>F</sub>
	L		+10V	X	X	↑	X	X	Q <sub>P</sub>	L	L <sup>9</sup>
	L		+10V	X	X	↑	X	X	Q <sub>P</sub>	H	H <sup>9</sup>
	L		X	+10V	X	↑	X	X	L	Q <sub>F</sub>	L <sup>9</sup>
	L		X	+10V	X	↑	X	X	H	Q <sub>F</sub>	H <sup>9</sup>
	L		X	X	+10V	X	X	X	Q <sub>P</sub>	Q <sub>F</sub>	Q <sub>P</sub>
	L		X	X	X	X	X	X	Q <sub>P</sub>	Q <sub>F</sub>	Q <sub>F</sub>
	L		X	X	X	X	X	X	Q <sub>P</sub>	Q <sub>F</sub>	Q <sub>F</sub>
	■	H	X	X	*	X	X	X	Q <sub>P</sub>	Q <sub>F</sub>	Hi - Z
		X	+10V	X	X	↑	X	X	Q <sub>P</sub>	L	L <sup>9</sup>
		X	+10V	X	X	↑	X	X	Q <sub>P</sub>	H	H <sup>9</sup>
		X	X	+10V	X	↑	X	X	L	Q <sub>F</sub>	L <sup>9</sup>
		X	X	+10V	X	↑	X	X	H	Q <sub>F</sub>	H <sup>9</sup>
		L	X	X	+10V	X	X	X	Q <sub>P</sub>	Q <sub>F</sub>	Q <sub>P</sub>
		L	X	X	X	X	X	X	Q <sub>P</sub>	Q <sub>F</sub>	Q <sub>F</sub>
		L	X	X	X	X	X	X	Q <sub>P</sub>	Q <sub>F</sub>	Q <sub>F</sub>
	■	L	X	X	X	↑	L	L	Q <sub>P</sub>	Q <sub>F</sub>	Q <sub>F</sub>
		L	X	X	X	↑	L	H	L	L	L
		L	X	X	X	↑	H	L	H	H	H
		L	X	X	X	↑	H	H	H	H	H
	↑	X	X	X	X	X	X	X	X	X	H

**NOTES:**

- Positive Logic:  
 $S/R$  (or  $J/K$ ) =  $T_0 + T_1 + T_2 + \dots + T_{63}$   
 $T_n = (C_0, C_1)(I_0, I_1, I_2, \dots) (P_0, P_1, \dots, P_7)$
- Either Initialization (Active - High) or Output Enable (Active - Low) are available, but not both. The desired function is a user-programmable option.
- ↑ denotes transition from Low-to-High level.
- \* = H/L/+10V
- X = Don't Care ( $\leq 5.5V$ )
- H/L implies that either a High or a Low can occur, depending upon user programmed selection (each State and Output Register individually programmable).
- When using the  $F_n$  pins as inputs to the State and Output Registers in diagnostic mode, the  $F$  buffers are tri-stated and the indicated levels on the output pins are forced by the user.

**VIRGIN STATE**

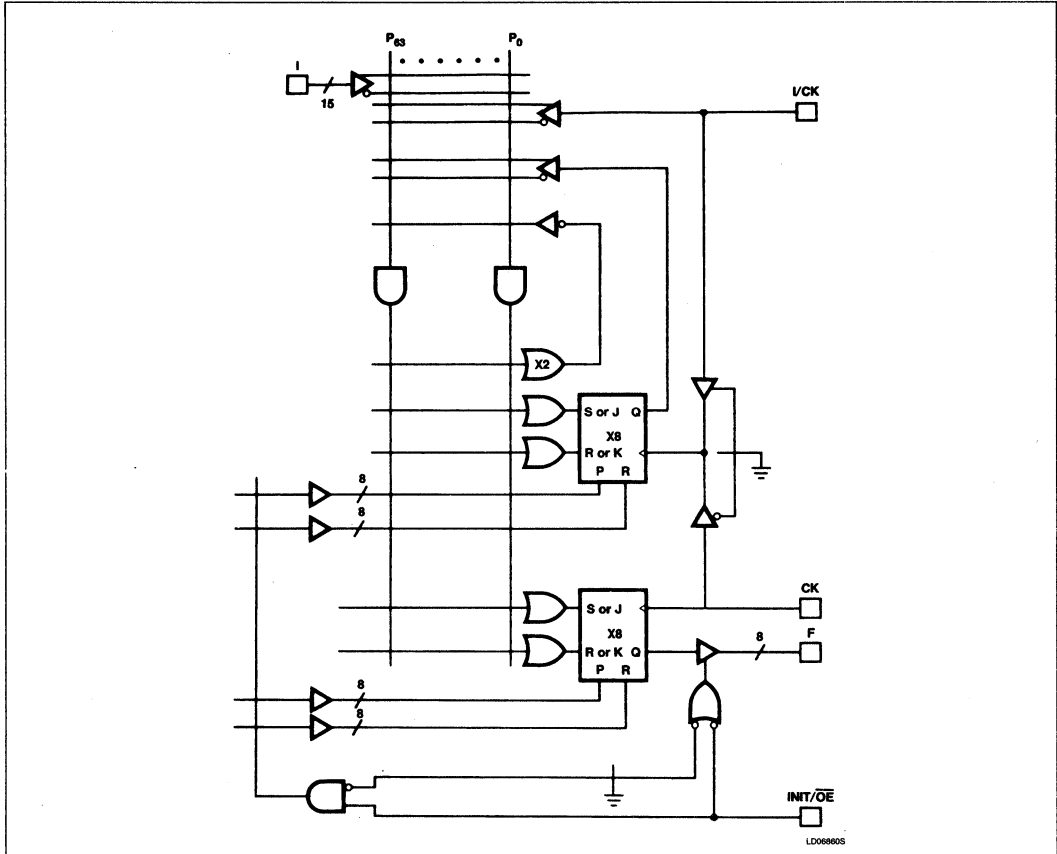
A factory-shipped virgin device contains all fusible links intact, such that:

- INIT/ $\overline{OE}$  option is set to INIT. In order to use the INIT function, the user must select either the PRESET or the RESET option for each flip-flop. Note that regardless of the user programmed initialization, or even if the INIT function is not used, all registers are preset to "1" by the power-up procedure.
- All transition terms are disabled (0).
- All S/R (or J/K) flip-flop inputs are disabled (0).
- The device can be clocked via a Test Array preprogrammed with a standard test pattern. NOTE: The Test Array pattern must be deleted before incorporating a user program.

# Field-Programmable Logic Sequencer (16 × 64 × 8)

PLUS405A

## FUNCTIONAL DIAGRAM

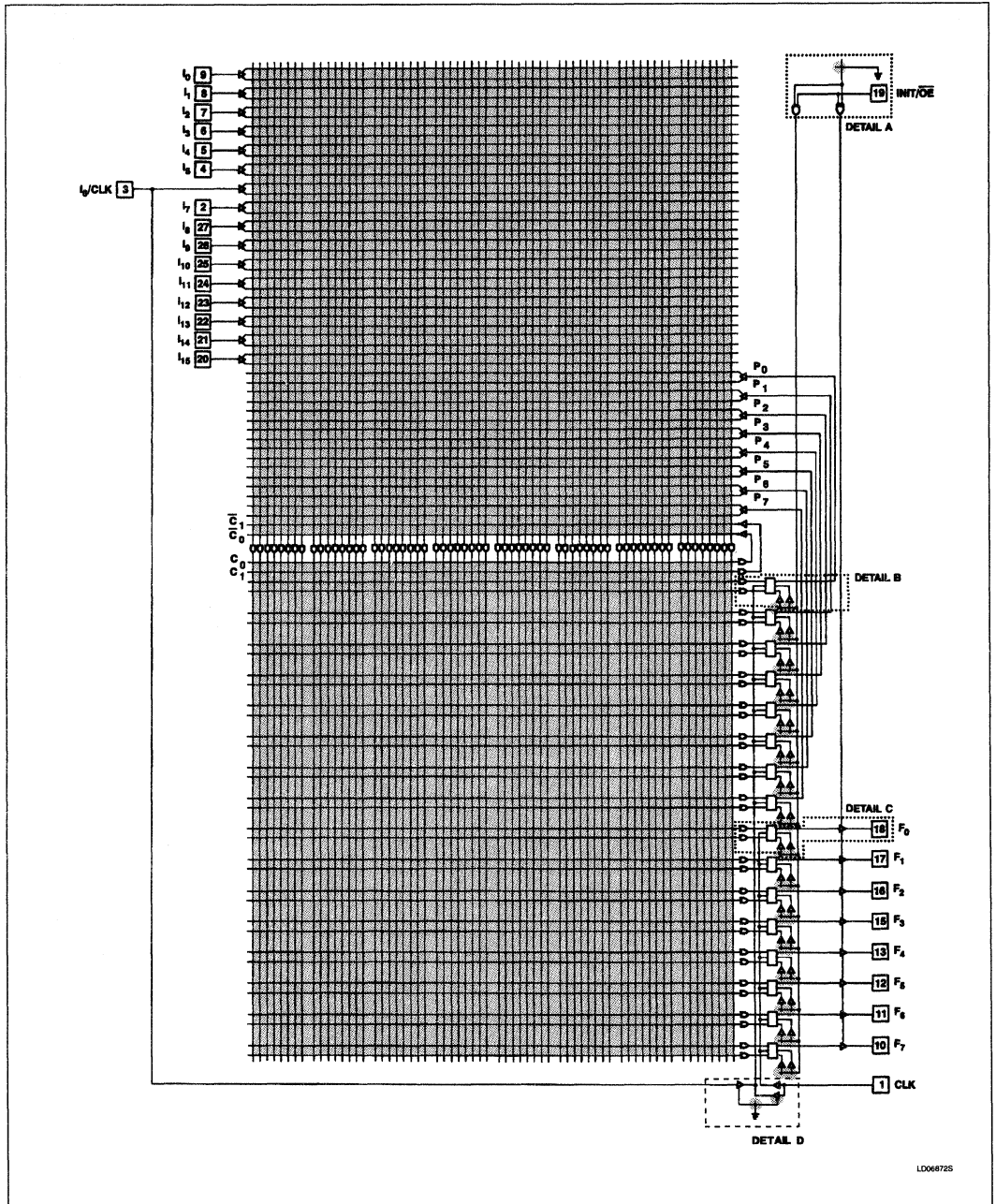


6

# Field-Programmable Logic Sequencer (16 × 64 × 8)

PLUS405A

## FPLS LOGIC DIAGRAM

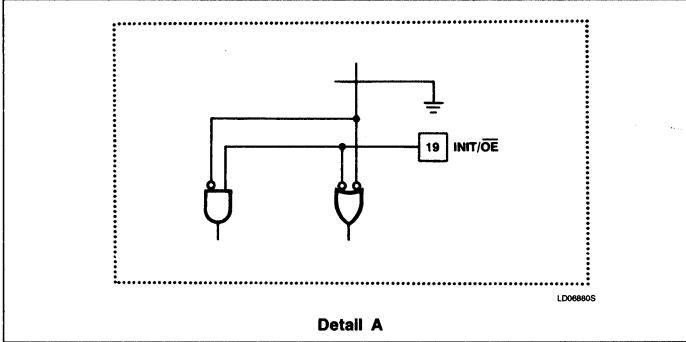


L0068725

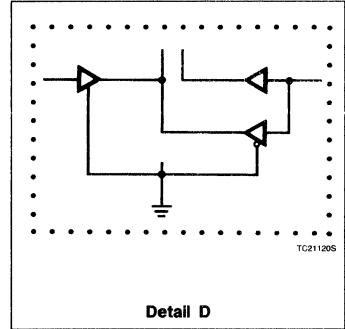
# Field-Programmable Logic Sequencer (16 × 64 × 8)

## PLUS405A

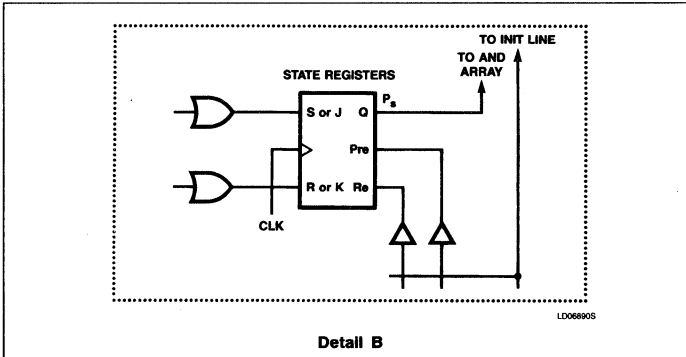
### DETAILS FOR REGISTERS FOR PLUS405A



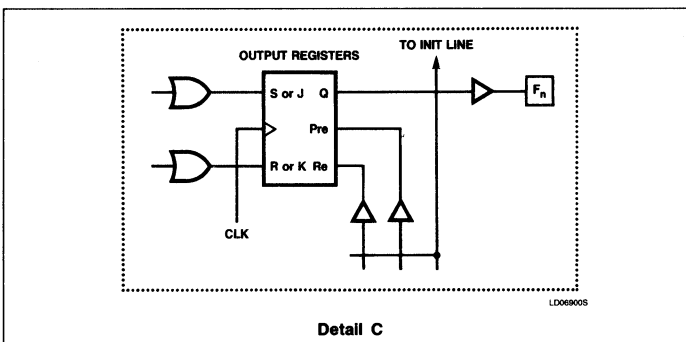
Detail A



Detail D



Detail B



Detail C

# Field-Programmable Logic Sequencer (16 × 64 × 8)

PLUS405A

## ORDERING INFORMATION

DESCRIPTION	ORDER CODE
28-Pin Plastic DIP 600mil-wide	PLUS405AN
28-Pin Plastic Leaded Chip Carrier	PLUS405AA

## THERMAL RATINGS

TEMPERATURE	
Maximum junction	150°C
Maximum ambient	75°C
Allowable thermal rise ambient to junction	75°C

ABSOLUTE MAXIMUM RATINGS<sup>1</sup>

SYMBOL	PARAMETER	RATING		UNIT
		Min	Max	
V <sub>CC</sub>	Supply voltage		+7	V <sub>DC</sub>
V <sub>IN</sub>	Input voltage		+5.5	V <sub>DC</sub>
V <sub>OUT</sub>	Output voltage		+5.5	V <sub>DC</sub>
I <sub>IN</sub>	Input currents	-30	+30	mA
I <sub>OUT</sub>	Output currents		+100	mA
T <sub>A</sub> T <sub>STG</sub>	Temperature range Operating Storage	0 -65	+75 +150	°C

DC ELECTRICAL CHARACTERISTICS 0°C ≤ T<sub>A</sub> ≤ +75°C, 4.75 ≤ V<sub>CC</sub> ≤ 5.25V

SYMBOL	PARAMETER	TEST CONDITIONS	LIMITS			UNIT
			Min	Typ <sup>2</sup>	Max	
<b>Input voltage<sup>3</sup></b>						
V <sub>IH</sub> V <sub>IL</sub> V <sub>IC</sub>	High Low Clamp <sup>4</sup>	V <sub>CC</sub> = Max V <sub>CC</sub> = Min V <sub>CC</sub> = Min, I <sub>IN</sub> = -12mA	2		0.8 -1.2	V
<b>Output voltage<sup>3</sup></b>						
V <sub>OH</sub> V <sub>OL</sub>	High Low	V <sub>CC</sub> = Min I <sub>OH</sub> = -2mA I <sub>OL</sub> = 9.6mA	2.4	0.35	0.45	V
<b>Input current</b>						
I <sub>IH</sub> I <sub>IL</sub> I <sub>IL</sub>	High Low Low (CK input)	V <sub>IN</sub> = 5.5V V <sub>IN</sub> = 0.45V V <sub>IN</sub> = 0.45V		< 1 -10 -50	25 -100 -250	μA
<b>Output current</b>						
I <sub>O(OFF)</sub> I <sub>OS</sub>	Hi-Z state Short circuit <sup>4,5</sup>	V <sub>CC</sub> = Max V <sub>OUT</sub> = 5.5V V <sub>OUT</sub> = 0.45V V <sub>OUT</sub> = 0V		1 -1	40 -40 -70	μA mA
I <sub>CC</sub>	V <sub>CC</sub> supply current <sup>6</sup>	V <sub>CC</sub> = Max		190	225	mA
<b>Capacitance</b>						
C <sub>IN</sub> C <sub>OUT</sub>	Input Output	V <sub>CC</sub> = 5.0V V <sub>IN</sub> = 2.0V V <sub>OUT</sub> = 2.0V		8 10		pF

## NOTES:

- Stresses above those listed under "Absolute Maximum Ratings" may cause malfunction or permanent damage to the device. This is a stress rating only. Functional operation at these or any other condition above those indicated in the operational and programming specification of the device is not implied.
- All typical values are at V<sub>CC</sub> = 5V, T<sub>A</sub> = +25°C.
- All voltage values are with respect to network ground terminal.
- Test one at a time.
- Duration of short circuit should not exceed 1 second.
- I<sub>CC</sub> is measured with the INIT/OE input grounded, all other inputs at 4.5V and the outputs open.

# Field-Programmable Logic Sequencer (16 × 64 × 8)

PLUS405A

**AC ELECTRICAL CHARACTERISTICS**  $R_1 = 470\Omega$ ,  $R_2 = 1k\Omega$ ,  $C_L = 30pF$ ,  $0^\circ C \leq T_A \leq +75^\circ C$ ,  $4.75V \leq V_{CC} \leq 5.25V$ 

SYMBOL	PARAMETERS	TO	FROM	LIMITS			UNIT
				Min	Typ <sup>1</sup>	Max	
<b>Pulse width</b>							
$t_{CKH}$	Clock high	CK-	CK+	10	8		ns
$t_{CKL}$	Clock low	CK+	CK-	10	8		
$t_{CKP1}$	Period (without Complement Array)	CK+	CK-	20	15		
$t_{CKP2}$	Period (with Complement Array)	CK+	CK+	30	25		
$t_{INITH}$	Initialization pulse	INIT+	INIT-	10	8		
<b>Setup time<sup>2</sup></b>							
$t_{IS1}$	Input	CK+	Input±	15	12		ns
$t_{IS2}$	Input (through Complement Array)	CK+	Input±	25	20		
$t_{VS}$	Power-on preset	CK-	$V_{CC+}$	0	-10		
$t_{INITS}$	Initialization	CK-	INIT-	0	-10		
<b>Hold time</b>							
$t_{IH}$	Input	Input±	CK+	0	-5		ns
<b>Propagation delay</b>							
$t_{CKO}$	Clock	Output±	CK+		8	10	ns
$t_{OE}$	Output enable	Output-	OE-		12	15	
$t_{OD}$	Output disable	Output+	OE+		12	15	
$t_{INIT}$	Initialization	Output+	INIT+		15	20	
$t_{PPR}$	Power-on preset	Output+	$V_{CC+}$		0	10	
<b>Frequency of operation</b>							
$f_{MAX}$	Without Complement Array					40	MHz
$f_{MAX}$	With Complement Array					28.5	

**NOTES:**

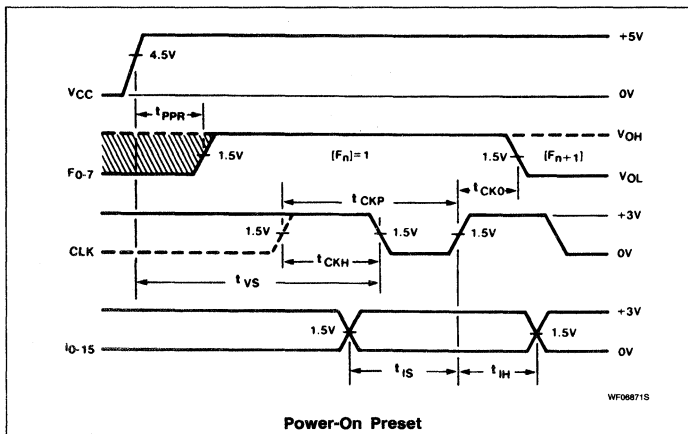
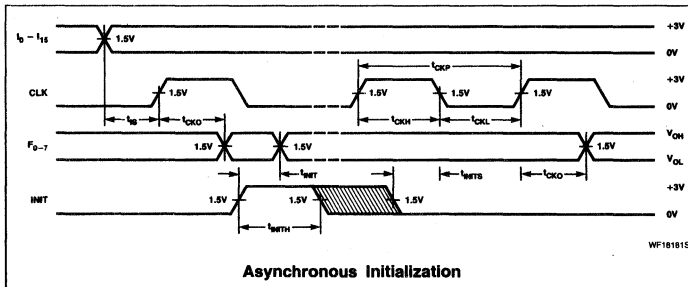
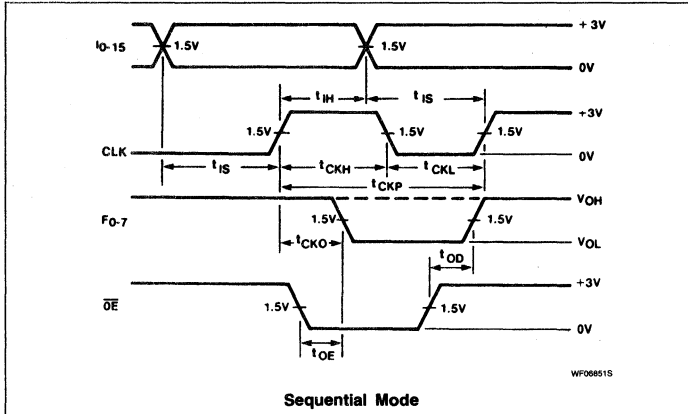
- All typical values are at  $V_{CC} = 5V$ ,  $T_A = +25^\circ C$ .
- Product term loading considerations: Connecting more than 8 toggle-mode registers to only one product term by itself may compromise the setup time performance. Product term loading restrictions do not apply if toggle mode functions are not used.



# Field-Programmable Logic Sequencer (16 × 64 × 8)

## PLUS405A

### TIMING DIAGRAMS



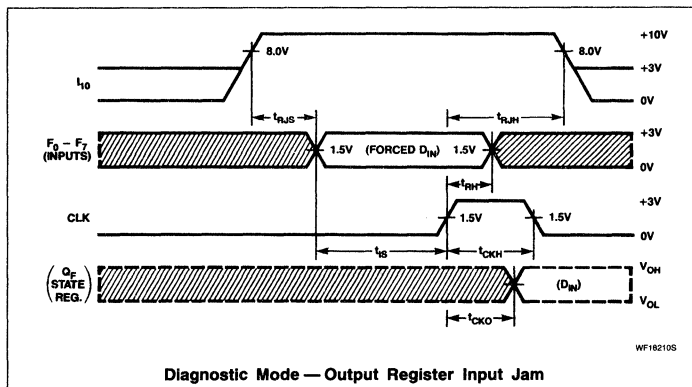
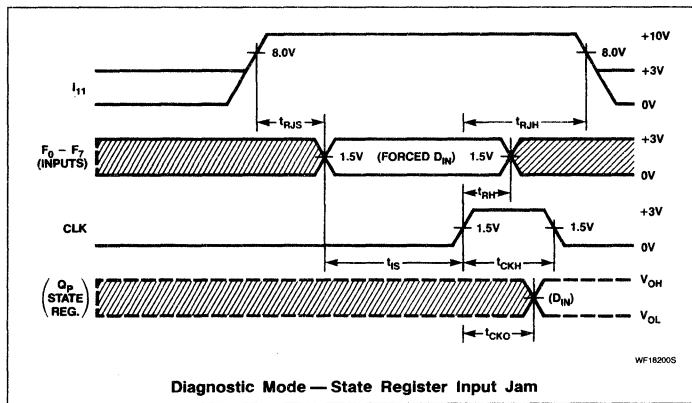
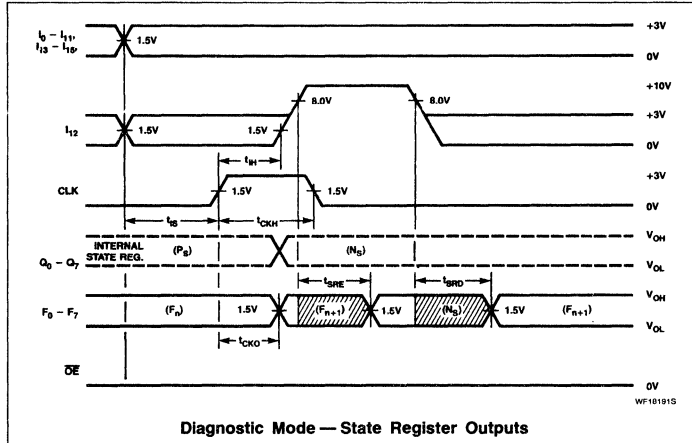
### TIMING DEFINITIONS

- $t_{CKH}$  Width of input clock pulse.
- $t_{CKL}$  Interval between clock pulses.
- $t_{CKP}$  Clock period — when not using Complement Array.
- $t_{IS1}$  Required delay between beginning of valid input and positive transition of Clock.
- $t_{CKP2}$  Clock period — when using Complement Array.
- $t_{IS2}$  Required delay between beginning of valid input and positive transition of Clock, when using optional Complement Array (two passes necessary through the AND Array).
- $t_{VS}$  Required delay between  $V_{CC}$  (after power-on) and negative transition of Clock preceding first reliable clock pulse.
- $t_{INITS}$  Required delay between negative transition of Asynchronous Initialization and negative transition of Clock preceding first reliable clock pulse.
- $t_{IH}$  Required delay between positive transition of Clock and end of valid Input data.
- $t_{CKO}$  Delay between positive transition of Clock and when Outputs become valid (with INIT/OE low).
- $t_{OE}$  Delay between beginning of Output Enable Low and when Outputs become valid.
- $t_{OD}$  Delay between beginning of Output Enable High and when Outputs are in the off state.
- $t_{SRE}$  Delay between input  $I_{12}$  transition to Diagnostic Mode and when the Outputs reflect the contents of the State Register.
- $t_{SRD}$  Delay between input  $I_{12}$  transition to Logic mode and when the Outputs reflect the contents of the Output Register.
- $t_{INIT}$  Delay between positive transition of Initialization and when Outputs become valid.
- $t_{PPR}$  Delay between  $V_{CC}$  (after power-on) and when Outputs become preset at "1".
- $t_{INITH}$  Width of initialization input pulse.
- $t_{RH}$  Required delay between positive transition of Clock and end of valid Input data when jamming data into State or Output Registers in diagnostic mode.
- $t_{RJS}$  Required delay between inputs  $I_{11}$  or  $I_{10}$  transition to State and Output Register Input Jam Diagnostic Modes, respectively, and when the output pins become available as inputs.
- $t_{RJH}$  Required delay between positive transition of Clock and end of inputs  $I_{11}$  or  $I_{10}$  transition to State and Output Register Input Jam Diagnostic Modes, respectively.
- $f_{MAX}$  Maximum clock frequency.

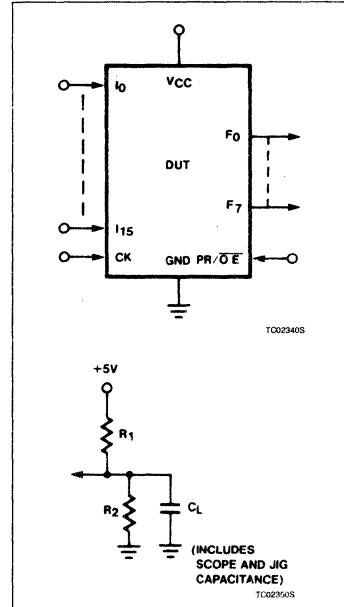
# Field-Programmable Logic Sequencer (16 × 64 × 8)

PLUS405A

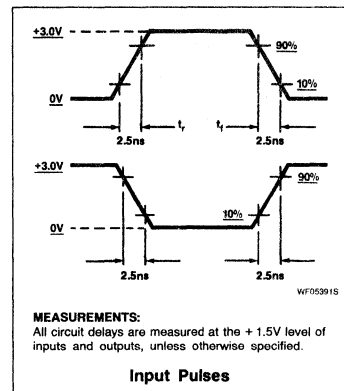
## TIMING DIAGRAMS (Continued)



## TEST LOAD CIRCUIT



## VOLTAGE WAVEFORM



# Field-Programmable Logic Sequencer (16 × 64 × 8)

PLUS405A

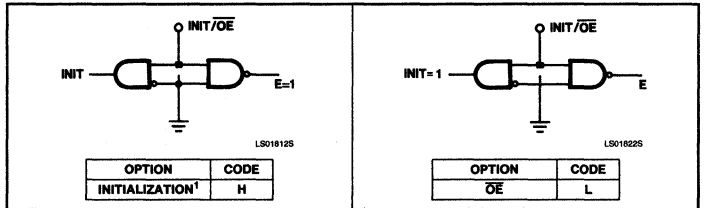
## LOGIC PROGRAMMING

The FPLS can be programmed by means of Logic programming equipment.

With Logic programming, the AND/OR gate input connections necessary to implement the desired logic function are coded directly from the State Diagram using the Program Table of the following page.

In this table, the logic state or action of control variables C, I, P, N, and F, associated with each Transition Term  $T_n$ , is assigned a symbol which results in the proper fusing pattern of corresponding link pairs, defined as follows:

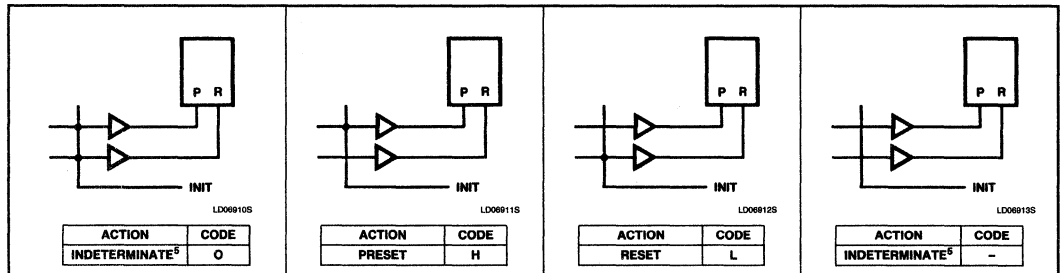
## INITIALIZATION/OE OPTION - (IN/E)



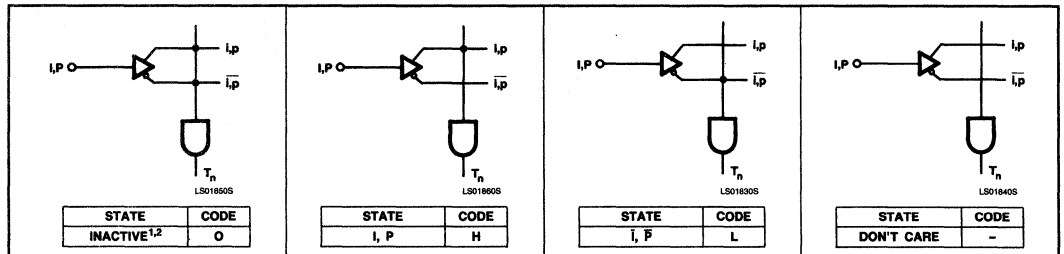
### PROGRAMMING THE PLUS405A:

The PLUS405A has a power-up preset feature. This feature insures that the device will power-up in a known state with all register elements (State and Output Register) at a logic High (H). When programming the device it is important to realize this is the initial state of the device. You *must* provide a next state jump if you do not wish to use all Highs (H) as the present state.

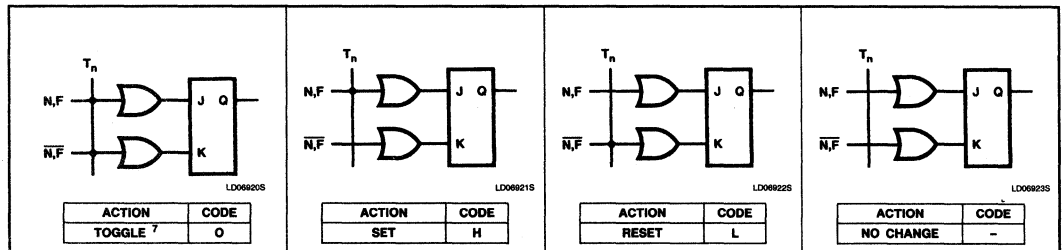
## PRESET/RESET OPTION - (P/R)



## "AND" ARRAY - (I), (P)



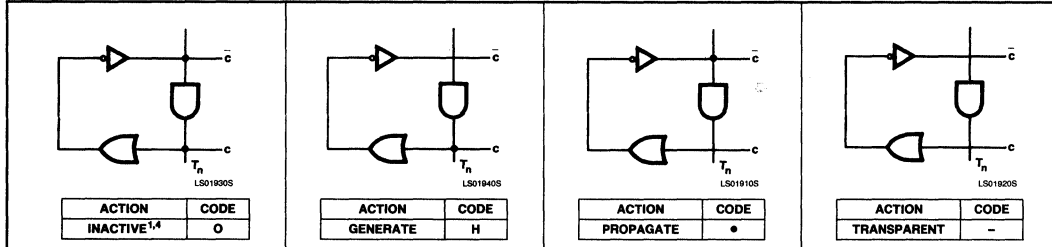
## "OR" ARRAY - J-K FUNCTION - (N), (F)



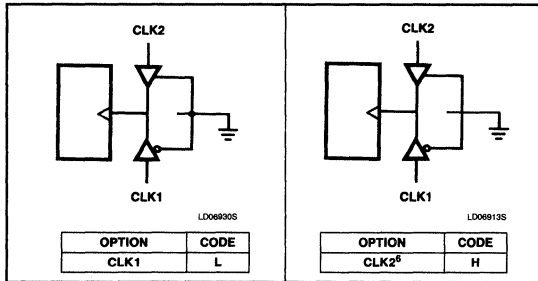
# Field-Programmable Logic Sequencer (16 × 64 × 8)

PLUS405A

## "COMPLEMENT" ARRAY - (C)



## CLOCK OPTION - (CLK1/CLK2)



**NOTES:**

1. This is the initial unprogrammed state of all links. It is normally associated with all unused (inactive) AND gates  $T_n$ .
2. Any gate  $T_n$  will be unconditionally inhibited if any one of its I or P link pairs is left intact.
3. To prevent simultaneous Set and Reset flip-flop commands, this state is not allowed for N and F link pairs coupled to active gates  $T_n$  (see flip-flop truth tables).
4. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates  $T_n$ .
5. These states are not allowed when using PRESET/RESET option.
6. Input buffer  $I_6$  must be deleted from the AND array (i.e. all fuse locations "Don't Care") when using second clock option.
7. A single product term cannot drive more than 8 registers by itself when used in TOGGLE mode.

6

# Field-Programmable Logic Sequencer (16 × 64 × 8)

PLUS405A

## PLUS405A FPLS PROGRAM TABLE

CUSTOMER NAME _____ SIGNETICS DEVICE # _____ CF (XXXX) CUSTOMER SYMBOLIZED PART # _____ REV _____ DATE _____ PROGRAM TABLE _____		<b>AND</b> INACTIVE 0 GENERATE A PROPAGATE 0 TRANSPARENT - INACTIVE 0 I, P H I, F L DON'T CARE -	<b>OR</b> INACTIVE OR TOGGLE 0 SET H RESET L NO CHANGE - INACTIVE OR TOGGLE 0 SET H RESET L NO CHANGE - <b>OPTION</b> INIT H OE L INIT/OE
<b>AND</b>		<b>OR</b>	
INPUT (Im)		PRESENT STATE (Ps)	
C01 C0	I14 I13 I12 I11 I10 I9 I8 I7 I6 I5 I4 I3 I2 I1 I0	P7 P6 P5 P4 P3 P2 P1 P0	N7 N6 N5 N4 N3 N2 N1 N0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63	F7 F6 F5 F4 F3 F2 F1 F0	INITIALIZATION/OUTPUT ENABLE	
CLOCK 1/2		OR	
CONN ARRAY		NEXT STATE (Ns)	
PIN NO.		PIN LABELS	

**NOTES:**

1. The FPLS is shipped with all links initially intact. Thus, a background of "0" for all Terms, and an "H" for the IN/E and H for the clock option, exits in the table, shown BLANK instead for clarity.
2. Unused C<sub>n</sub>, I<sub>m</sub>, and P<sub>s</sub> bits are normally programmed Don't Care (-).
3. Unused Transition Terms can be left blank for future code modification, or programmed as (-) for maximum speed.

TB031415

Effective October 1986, this section has been superseded by the Military Products Data Manual. Information regarding this manual can be obtained from the Military Division in Sacramento, (916) 925-6700. Electrical specifications herein do not necessarily describe the performance characterization of military processed products.



**Signetics**

**Section 8**

**Application Specific Products**

**INDEX**

<b>Section 8 - Development Software</b>	
AMAZE - Automatic Map and Zap Equations Design Software .....	8-3





### Application Specific Products

#### SOFTWARE SUPPORT FOR USER PROGRAMMABLE LOGIC

Computer Aided Design (CAD) support is becoming necessary to address the time-consuming details required by the more complex programmable logic devices available today. The design effort can include the manipulation of Boolean equations, truth tables, state diagrams, flowcharts etc, to create the binary fuse map required to program such devices.

For many years, design engineers have used programmable read-only memories (PROM's) to replace conventional logic. The architecture of the PROM can be described as a programmable logic device containing a **fixed** AND array followed by a **programmable** OR array. The fixed structure of the PROM requires a full assignment of output words to be programmed for every input combination. Therefore, little use can be made of programmable logic software programs for logic minimization or other compiling efforts when using PROM's.

Signetics Programmable Logic Devices are the most advanced approach to solving the inherent limitations of PROM's. Their architecture consists of a **programmable** AND array, followed by a **programmable** OR array - with the addition of a **programmable** invert function for flexible output control.

A Signetics PLD device can implement any set of Boolean expressions, provided that they are first put into the standard sum of products form. The logical AND's are implemented at the first gate level of the programmable logic device and the logical OR's are implemented by the second gate level within the PLD. The only limitations on the expressions are those imposed by the number of inputs, outputs, and internal product terms provided by the particular PLD circuit selected. The efficiency of implementing the set of equations can be increased significantly by applying DeMorgan's theorem, and utilizing the programmable invert function on each output.

If there seems to be too few product terms to handle a relatively large equation set, one of several minimization methods can be pursued.

The probability of reducing such equations to manageable size is enhanced through the flexibility of shared AND terms for each output function, the accessibility of all AND terms to each output, and having a programmable invert function on each output. All of these features can be utilized by applying the manual manipulation of Venn Diagrams or Karnaugh Maps. However, the time and effort to accomplish these tasks as well as document the effort for procure-

ment specification purposes increase the need and desire to have software programs to automatically perform such manipulations.

Many types of software programs are being developed to provide this assistance for operation on a wide range of computer hardware. This list of software is expanding rapidly, consisting of both Signetics generated software and some independent software houses' contributions.

This discussion is intended to outline the Signetics developed software program called AMAZE (Automatic Map And Zap Equation Entry) The AMAZE software program currently consists of five modules, BLAST ('Boolean Logic And State Transfer' entry program), PTP ('PAL To PLD' conversion program), DPI ('Device Programmer Interface' program), PLD SIM ('PLD Simulator' program) and the PTE ('Program Table Editor' program). Other modules will be added when product developments require additional software tools.

It must be noted that the AMAZE program is not by any means the total extent of software available for use in designing with PLD (Programmable Logic Devices). Please contact your local Signetics representative for the latest word on the most currently available software.

# Programmable Logic Development Software

# AMAZE

## AMAZE

### Description

Automatic Map And Zap Equation Entry software program. The AMAZE software program consists of the following five modules:

- **BLAST** ('Boolean Logic And State Transfer' entry program)
  - **PTE** ('Program Table Editor')
  - **PTP** ('PAL To PLD' conversion program)
  - **DPI** ('Device Programmer Interface' program)
  - **PLD SIM** ('PLD Simulator' program)
- Each module performs specific tasks as outlined in the following sections.

### Features

- Multiple modules allowing expansion for future requirements
- Each module designed to be user friendly
- Both HELP and ERROR messages
- Document printout: Header, Pin diagram, Boolean equation and Fuse map
- Interface with most commercial programmers
- SIMULATOR programs provide both test and applications assistance

### Equipment

- The PC version of AMAZE is capable of operating on an IBM-PC with MS-DOS operating system and having 512K bytes of on-board RAM
  - The VAX version of AMAZE is capable of operating on a VAX computer with VMS operating system
- Please contact your Signetics Sales Representative for availability and ordering information.

## Products Supported

AMAZE will support the following products:

- **20-pin PLDs:** PLS151  
PLS153  
PLS153A  
PLHS153  
PLUS153  
PLS155  
PLS157  
PLS159  
PLS159A  
PLHS18P8A
- **24-pin PLDs:** PLS161  
PLS162  
PLS163  
PLS167  
PLS167A  
PLS168  
PLS168A  
PLS173  
PLUS173  
PLS179  
PLHS473  
PLC473
- **28-pin PLDs:** PLS100  
PLS101  
PLS103  
PLS105  
PLS105A  
PLC105  
PLUS405A
- **52-pin PLDs:** PLHS501  
PLHS502

# Programmable Logic Development Software

# AMAZE

## BLAST

**Boolean Logic And State Transfer** program is a menu driven software package that supports the engineer in implementing logic designs into Signetics Programmable Logic Devices. It checks design data and automatically compiles a program table from Boolean and State Machine equations. Data from the program table is then used to produce a Standard File which contains the fusing codes in a form acceptable to all the AMAZE modules (i.e. PLD-SIM and DPI).

BLAST reports the logic and syntax errors, and lists the equations in a Sum of Products form, which can help the user to minimize the entered logic equations. It will automatically partition State machine designs into specified devices, and then delete redundant terms during compilation.

BLAST also provides the capability of modifying a current logic set programmed into a device by overlaying new data onto unused fuses.

## BOOLEAN LOGIC AND STATE TRANSFER FEATURES

- User friendly interactive pinlist editor
- Boolean equation or state vector entry
- Schematic entry (with external schematic capture package)\*\*
- On-line error checking, minimization, and design overlay
- Capable of partitioning single designs into multiple PLDs
- Generates standard PLD fusemap files
- Supports all Signetics' PLDs and PLEs
- User definable device files for support of PALs and other PLD devices

## PTE

Program Table Entry is an interactive editor which allows the logic designer to enter data into AMAZE in the form of SIGNETICS APPROVED PROGRAM TABLES. Each Signetics PLD data sheet has the program table format which applies to that device. In addition, PTE can be used to document completed designs and to make changes in logic functions which have been previously defined in the BLAST module.

## PROGRAM TABLE EDITOR FEATURES

- Allows easy creation and editing of new and existing PLD designs
- Truth-table representation of PLD fusemap in High/Low format
- On-line editor provides automatic cursor control and prevents syntax errors
- On-line help screen and print facility
- Operates on standard PLD fusemap files

## PTP

PAL<sup>TM</sup> To PLD is a conversion program to allow easy transfer of the various PAL 20 and 24 pin circuits to the Signetics PLD 20 and 24 pin series devices.

PTP can automatically upload the PAL<sup>TM</sup> pattern from a Commercial programmer, convert the pattern into a PLD pattern, and then download the PLD pattern into the programmer. The PAL<sup>TM</sup> pattern and it's corresponding PLD pattern are documented, and the PLD pattern can be directed to other AMAZE modules.

PTP can also convert the PAL<sup>TM</sup> fuse file in a HEXPLOT format.

## PAL-TO-PLD CONVERSION FEATURES

- Menu-driven fusemap conversion of 20- and 24-pin PALs to pin and functional equivalent Signetics' PLDs
- Automatic assembler removes duplicated p-terms providing efficient PLD mapping
- Accepts JEDEC, fuseplot files or direct PAL master input via commercial PLD programmer
- User selectable RS-232 programmable interface parameters
- Provides fusemap conversion documentation
- Generates standard PLD fusemap files compatible with other AMAZE modules

## DPI

Device Programmer Interface is the software module that provides the interface between the Standard File created by the AMAZE modules and a commercial programmer. This

module allows both download (Sending from Host to programmer) and upload (Sending from programmer into the host) operations.

DPI supports both JEDEC and Signetics H & L formats to convey fusing information to and from several commercial programmers.

## DEVICE PROGRAMMER INTERFACE FEATURES

- Supports standard JEDEC and Signetics High/Low fusemap file formats
- RS-232 interface to commercial PLD programmers
- Screen menus for easy upload and download of fusemaps
- User selectable RS-232 parameters for programmer flexibility
- Test vectors automatically transferred to programmer along with fusemap file
- Operates with standard PLD fusemap files

## PLD SIMULATOR

The PLD Simulator program is a software package that simulates the operation of the logic that has been defined for Signetics PLD products. The input to the program is the Standard File generated by other AMAZE modules. The simulator has the capability of running manually or automatically. In the automatic mode the simulator creates a file of test vectors that can be used to test the programmed devices. In the manual mode the program will allow the operator to assign an input vector and observe the resultant output.

## PLD FUNCTIONAL SIMULATOR FEATURES

- Functional simulation of designs created from equations, program tables, or existing programmed devices
- Automatic test vector generation from standard or JEDEC PLD fusemap files
- Interactive keyboard entry or batch file input of test vectors
- Detects Illegal State Machine transitions and flags affected p-terms
- On-line help screen

\*PAL<sup>TM</sup> and PLE<sup>TM</sup> are trademarks of Monolithic Memories, Inc.  
 (\*\* AMAZE accepts TTL schematics generated with DATA-I/O FUTURENET DASH)

## Programmable Logic Development Software

AMAZE

---

### **AMAZE HARDWARE REQUIREMENTS**

P/N

- 82SOFT523-SS IBM-PC or IBM-XT or compatible computer  
512K Memory  
PC-DOS Operating System Version 2.0 or higher.  
2 Floppy Disk Drives or  
1 Floppy and 1 Hard disk
- 82SOFT211-SS VAX Series Computer  
VMS operating system –  
all revisions.

AMAZE comes fully documented with appropriate magnetic media. Applications support is provided by Signetics Field Application Engineers in most areas. Contact the Sales Office in your area for information on AMAZE and Signetics Programmable Logic Products.

### INDEX

#### Section 9 - Application Notes

<b>AN7</b>	Single Chip Multiprocessor Arbiter .....	9-3
<b>AN8</b>	Introduction to Signetics Programmable Logic .....	9-15
<b>AN10</b>	4-Bit Binary-To-7 Segment Decoder .....	9-27
<b>AN11</b>	PLD Programmable Retriggerable One-Shot .....	9-51
<b>AN12</b>	Low Cost Programmer for PLD 20-Pin Series .....	9-55
<b>AN13</b>	Oscillator With PLS159 .....	9-71
<b>AN14</b>	Latches and Flip-Flops With PLS153 .....	9-76
<b>AN15</b>	PLS159 Primer .....	9-89
<b>AN16</b>	PLS159 as Modulo Counters .....	9-100
<b>AN17</b>	Electronic Combination Lock .....	9-107
<b>AN18</b>	Schmitt Trigger Using PLS153 and PLS159 .....	9-110
<b>AN19</b>	Electronic Dice .....	9-118
<b>AN20</b>	Single Pin Debouncer Using PLS153/159 .....	9-122
<b>AN21</b>	9-Bit Parity Generator/Checker With PLS153/153A .....	9-125
<b>AN22</b>	PLS151 Toolkit .....	9-131
<b>AN23</b>	PLS168/168A Primer .....	9-139
<b>AN24</b>	PLS173 as a 10-Bit Comparator, 74LS460 .....	9-152
<b>AN25</b>	A Simple Entry/Exit Meeting Device (PLS157) .....	9-158
<b>AN26</b>	PLHS18P8A Primer .....	9-164
<b>AN27</b>	PLHS473 Primer .....	9-172
<b>AN28</b>	High-Speed 12-Bit Tracking A/D Converter Using PLS179 .....	9-180
<b>AN29</b>	PLHS501 Programmable Macro Logic Primer .....	9-190



## AN7 Single Chip Multiprocessor Arbiter

Application Note

### Application Specific Products

#### INTRODUCTION

In multiprocessor environments there is considerable savings to be made through sharing system resources. If each processor must support its own bus structure, I/O devices, and bulk storage medium, system cost could be very high. In the configuration shown in Figure 1, all processors share a common communication bus, and a number of system resources.

Since every processor must use the common system bus to communicate with its peripherals, a priority structure that resolves simultaneous processor bus requests into a single bus grant must be integrated into the system. In addition to making request-grant transactions, transient bus contention due to grant switching must be removed by inserting precise guard band times between bus grants.

Signetics' Field Programmable Logic Sequencer provides a convenient and cost-effective means for implementing a synchronous arbiter to perform these tasks within a single chip.

#### ARBITER STRUCTURE

Within a multiprocessor system, two general classes of processors can be recognized: Priority A and Priority b. Priority A processors have the highest request priority and must only compete with other Priority A processors for bus control. The arbiter must issue "A" grants in manner that prevents any high priority "A" processor from locking out another Priority A processor. To enable this, the Priority A rules implemented here use a Last Granted Lowest Priority (LGLP) ring structure. After an "A" processor has completed a bus-related task, its next arbitrated request priority will be lowest in the "A" request group. The previously second highest priority "A" processor will then become highest priority requester. The net effect of the "round robin" exchange is that every Priority A processor will have a turn at being highest priority processor. Priority A processors are typically ones that perform real-time operations or vital system tasks.

Priority b processors are lower in priority than the "A's" and may only be granted system control when no "A" requests are pending. "b" processors usually perform background tasks. Within the Priority b group, further priority ordering exists such that each "b" processor has a fixed priority position.

Plumber<sup>1</sup>, Pearce<sup>2</sup>, and Hojberg<sup>3</sup> present asynchronous techniques of arbiter implementation. These methods all have hard-wired priority rules and imprecise guard band times during grant switching. As pointed out by Hojberg, a synchronous state machine can be configured as a Mealy-type controller to provide not only precise guard band times and programmable priority rules, but also programmable input/output polarity. The state machine in Figure 2 is made from a control PROM array and an edge-triggered latch. The "A" and "b" requests and the machine's present state are used by the control PROM to determine the next "A" and "b" grants and the next state.

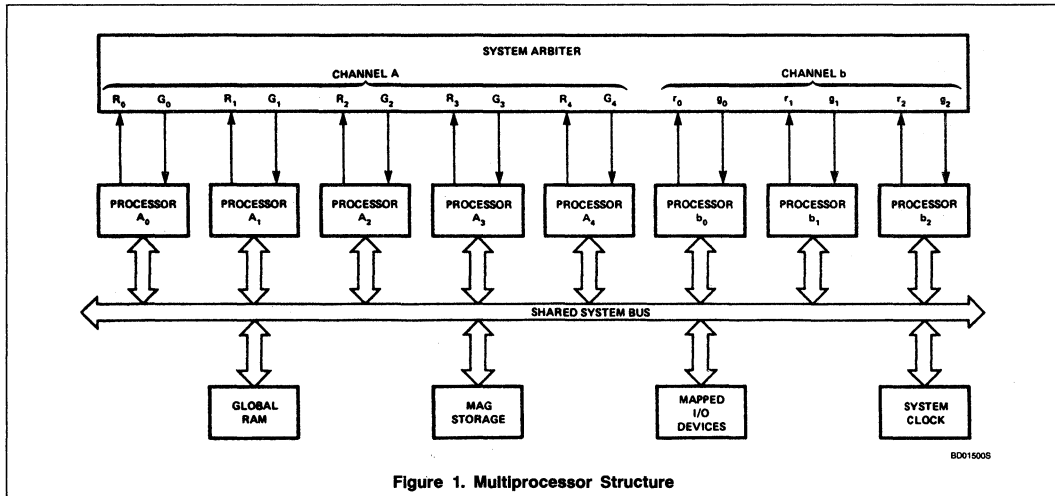
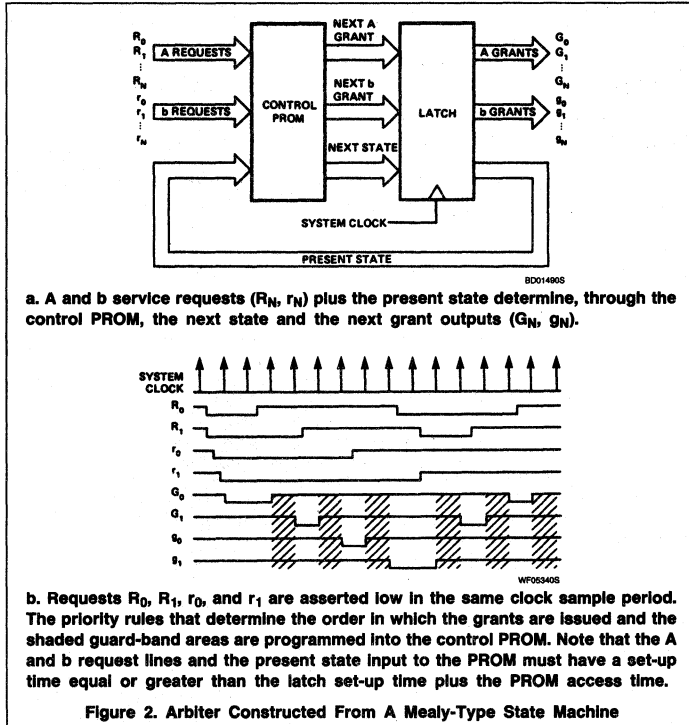


Figure 1. Multiprocessor Structure



# Single Chip Multiprocessor Arbitrer

AN7



## SYSTEM OPERATION

Two machine states can be identified by inspection: a wait state and a grant state. The state machine enters a grant state as a response to a system request on either  $R_N$  or  $r_N$ . The machine will remain in this state with a single grant line asserted as long as the request remains asserted. Upon releasing the request line, the machine will pass through a single wait state before considering other pending requests. This provides a single state guard band time. The requests received must meet the set-up requirement of the edge-triggered latch after propagating through the control PROM. If these time considerations do not fit within a given multiprocessor structure, an input latch may be added such that the  $R_N$  and  $r_N$  lines are clocked through the latch by the system clock, thereby removing asynchronous set-up time considerations. On the basis of a state machine approach, two techniques of implementation are feasible: 1) using an architecturally advanced single IC controller, the FPLS, and, 2) a traditional PROM/LATCH configuration.



# Single Chip Multiprocessor Arbitrer

AN7

The corresponding next state of each bit will be set to 0 for "L", 1 for "H", and No Change for ".". The FPLS's PR/OE line may be assigned either Asynchronous Preset or Output Enable functions, via a user programmable option.

The entire function is integrated into a single 28-pin package designated as PLS105.

### State Algorithm

Figure 5(a) displays the circular state form and all possible state transitions of the LGLP priority structure. Hex states 3F, 3E, 3D, 3C, and 3B are arbiter wait states  $W_{0-4}$ . In these states, processor "A" and "b" requests are monitored. Figure 5(b) illustrates a typical grant to processor  $A_1$  in hex state 07. As long as  $A_1$  asserts its request line low, the next state will be 07<sub>16</sub> and the next output will remain with  $G_1$  asserted low and all the other grant outputs asserted high. Since no change in state or grant output results from this transition, no PLA resources are required.

As soon as processor  $A_1$  returns its request line,  $R_1$ , to 1, a state transition is made to 3D, and an output transition is made to set all grant outputs to 1. Since processor  $A_1$  was the last to be granted system resources, it will now have the lowest A level request priority (LGLP). In wait state  $W_2$ , the highest priority processor will be  $A_2$ , second  $A_3$ , third  $A_4$ , and fourth  $A_0$ . To maintain the LGLP rule, grant transitions must follow the state rule  $G_N \rightarrow W_{(N+1)}$ , and wait states,  $W_M$ , must set their "A" priorities so that processor  $A_M$  is highest priority. Priority decreases as one proceeds clockwise around the state ring to the lowest priority processor,  $A_{(m-1)}$ .

When no "A" requests are pending, "b" requests may be granted. To avoid upsetting the LGLP priority rule, a "b" grant must leave and return to the same wait state. Since the "b" priority structure is the same regardless of the wait state, only a single set of "b" transition terms are required.

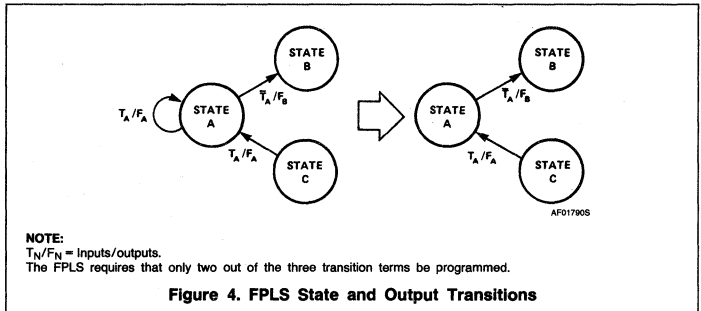


Figure 4. FPLS State and Output Transitions

For example, a grant transition to  $g_2$  (Hex 20-25) can be issued only if there are no "A", "b<sub>0</sub>", or "b<sub>1</sub>" requests pending. Given the binary wait state code 11XXXX, where "X's" represent Don't Cares, a request code of 01111111 will transfer the arbiter to the grant state  $g_2$  from any of the wait states,  $W_{0-4}$ .

It is important to realize that in making this transition, the lower three-state bits will not be changed—they provide the wait state return address. When  $r_2$  returns high, 1XXXXXXX, a transition back to the previously exited wait state is made by forcing a "1" in the three most significant state bits and leaving the lower three-state bits unchanged.

All output and state bits are initially preset to "1" through the use of the optional preset function. Grant output lines are only forced low when transitions are made to grant states and are returned to "1" when jumping back to a wait state.

Table 1 provides the complete arbiter program. The complete arbiter circuit diagram is shown in Figure 6. The AMAZE equations are shown in Figure 7.

### PROM/LATCH IMPLEMENTATION

The same five "A" processor and three "b" processor arbiter can be implemented with discrete PROM's and Latches using the same state diagrams for the FPLS, except that now looping transition terms must be programmed. Coding of all state and output transitions requires programming of two memory fields: the "A" request PROM's (2KX6) and the "b" request PROM (64 X 3). The complete circuit diagram is shown in Figure 6(b).

The "A" request PROM's determine the next machine state ( $N_{0-5}$ ) at all times, except when there are no "A" requests pending and there is a "b" request, or if the machine is presently in a "b" grant state. In these cases, the "b" request PROM controls the machine's next state.

The grant control lines are decoded from the next state lines and latched in two quad output latches. This PROM/LATCH organization is conceptually the same as that shown in Figure 2.

# Single Chip Multiprocessor Arbiter

AN7

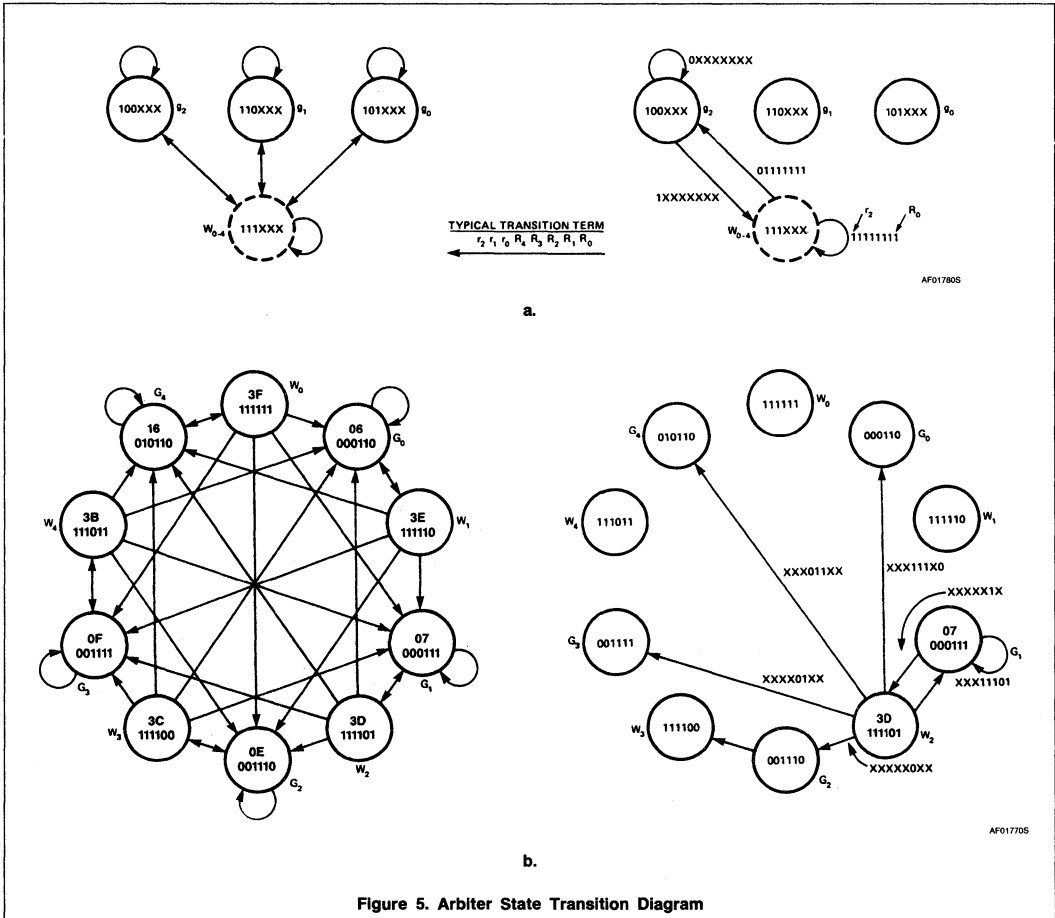
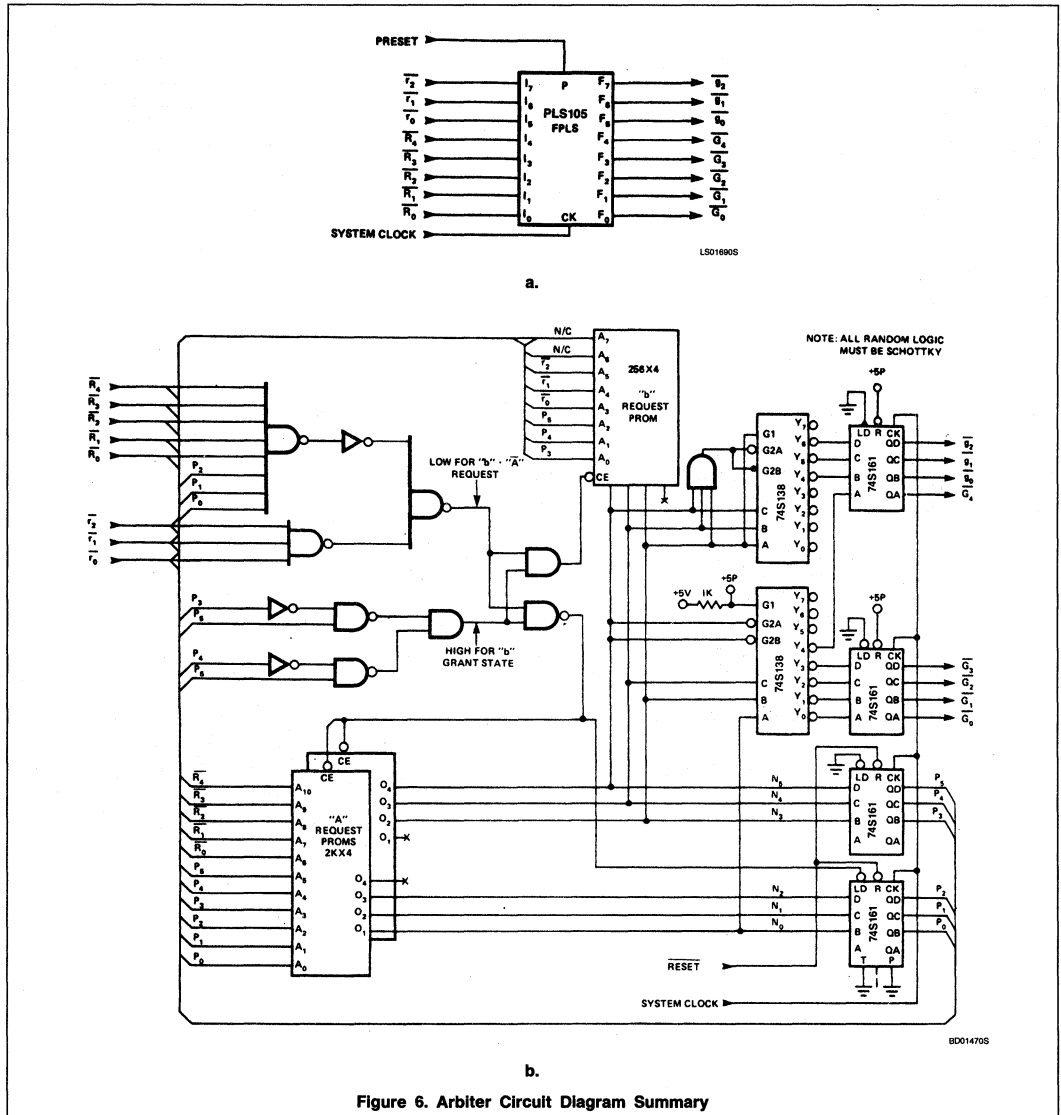


Figure 5. Arbiter State Transition Diagram

# Single Chip Multiprocessor Arbiter

AN7



# Single Chip Multiprocessor Arbiter

AN7

Table 1. FPLS Program Table for Priority Arbiter

TRANSITION TERM		INPUT VARIABLE (In)																PRESENT STATE (Ps)																OUTPUT TERM															
																																		NEXT STATE (Ns)								OUTPUT FUNCTION (Fn)							
No.		1	1	1	1	1	1	0	0	0	7	6	5	4	3	2	1	0	5	4	3	2	1	0	5	4	3	2	1	0	5	4	3	2	1	0	7	6	5	4	3	2	1	0					
0																		L	H	H	H	H	H	H	H	H	H	H	H	H	L	L	L	L	H	H	L	H	H	H	H	H	H	H	H	H	L		
1																		L	H	H	H	H	H	H	H	H	H	H	H	H	H	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	L	H	
2																		L	H	H	H	H	H	H	H	H	H	H	H	H	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	
3																		L	H	H	H	H	H	H	H	H	H	H	H	H	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H
4																		L	H	H	H	H	H	H	H	H	H	H	H	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	
5																		L	H	H	H	H	H	H	H	H	H	H	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H		
6																		L	H	H	H	H	H	H	H	H	H	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H			
7																		L	H	H	H	H	H	H	H	H	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H			
8																		L	H	H	H	H	H	H	H	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H			
9																		H	H	H	H	L	H	H	H	H	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H				
10																		L	H	H	H	H	H	H	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H					
11																		L	H	H	H	H	H	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H						
12																		L	H	H	H	H	H	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H						
13																		H	H	H	H	L	H	H	H	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	L	H							
14																		H	H	H	L	H	H	H	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	L	H								
15																		L	H	H	H	H	H	L	L	L	H	H	H	H	H	H	H	H	H	H	H	L	H										
16																		L	H	H	H	H	H	L	L	L	H	H	H	H	H	H	H	H	H	H	H	L	H										
17																		H	H	H	H	L	H	H	H	L	L	L	H	H	H	H	H	H	H	H	H	L	H										
18																		H	H	L	H	H	H	H	L	L	L	H	H	H	H	H	H	H	H	H	L	H											
19																		H	H	L	H	H	H	H	L	L	L	H	H	H	H	H	H	H	H	H	L	H											
20																		L	H	H	H	H	H	L	L	L	H	H	H	H	H	H	H	H	H	L	H												
21																		L	H	H	H	H	H	L	L	L	H	H	H	H	H	H	H	H	L	H													
22																		H	H	H	H	H	H	L	L	L	H	H	H	H	H	H	H	H	L	H													
23																		L	H	H	H	H	H	L	L	L	H	H	H	H	H	H	H	L	H														
24																		H	L	H	H	H	H	L	L	L	H	H	H	H	H	H	L	H															
25																		L	H	H	H	H	H	L	L	L	H	H	H	H	H	H	L	H															
26																		L	H	H	H	H	H	L	L	L	H	H	H	H	H	L	H																
27																		L	H	H	H	H	H	L	L	L	H	H	H	H	H	L	H																
28																		H	H	L	L	L	H	L	L	L	H	H	H	H	L	H																	
29																		H	H	L	L	L	H	L	L	L	H	H	H	H	L	H																	
30																		L	L	L	H	H	L	L	L	H	H	H	H	L	H																		
31																		H	H	L	L	L	H	L	L	L	H	H	H	H	L	H																	
32																		H	H	L	L	L	H	L	L	L	H	H	H	H	L	H																	
33																		H	L	H	H	H	L	L	L	H	H	H	H	L	H																		
34																		H	H	L	L	L	H	L	L	L	H	H	H	H	L	H																	
35																		H	L	L	L	H	L	L	L	H	H	H	H	L	H																		
36																		H	L	L	L	H	L	L	L	H	H	H	H	L	H																		

TS007865

## Single Chip Multiprocessor Arbiter

AN7

```

***** ARBITERS *****
@DEVICE SELECTION
ARBITERB/828105
@STATE VECTORS
[ FF5, FF4, FF3, FF2, FF1, FF0 ]
W0 = 03Fh ;
W1 = 03Eh ;
W2 = 03Dh ;
W3 = 03Ch ;
W4 = 03Bh ;
W04 = 111---b ;
GA0 = 06h ;
GA1 = 07h ;
GA2 = 0Eh ;
GA3 = 0Fh ;
GA4 = 16h ;
GB0 = 101---b ;
GB1 = 110---b ;
GB2 = 100---b ;

@INPUT VECTORS
@OUTPUT VECTORS
[QB2, QB1, QB0, QA4, QA3, QA2, QA1, QA0]
QA0' = Fh ;
QA1' = Fh ;
QA2' = Fh ;
QA3' = Fh ;
QA4' = Fh ;
QB0' = Fh ;
QB1' = Fh ;
QB2' = Fh ;
NOGRANT' = FFh ;

@TRANSITIONS
WHILE [W0]
CASE
  [/RA0] :: [GA0] WITH [QA0']
  [/RA1 * RA0] :: [GA1] WITH [QA1']
  [/RA2 * RA1 * RA0] :: [GA2] WITH [QA2']
  [/RA3 * RA2 * RA1 * RA0] :: [GA3] WITH [QA3']
  [/RA4 * RA3 * RA2 * RA1 * RA0] :: [GA4] WITH [QA4']
ENDCASE

WHILE [W1]
CASE
  [/RA1] :: [GA1] WITH [QA1']
  [/RA2 * RA1] :: [GA2] WITH [QA2']
  [/RA3 * RA2 * RA1] :: [GA3] WITH [QA3']
  [/RA4 * RA3 * RA2 * RA1] :: [GA4] WITH [QA4']
  [/RA0 * RA4 * RA3 * RA2 * RA1] :: [GA0] WITH [QA0']
ENDCASE

```

TB007506

## a. Arbiter State Equations

Figure 7

## Single Chip Multiprocessor Arbiter

AN7

```

WHILE [W2]
  CASE
    [/RA2] :: [GA2] WITH [QA2']
    [/RA3 * RA2] :: [GA3] WITH [QA3']
    [/RA4 * RA3 * RA2] :: [GA4] WITH [QA4']
    [/RA0 * RA4 * RA3 * RA2] :: [GA0] WITH [QA0']
    [/RA1 * RA0 * RA4 * RA3 * RA2] :: [GA1] WITH [QA1']
  ENDCASE

WHILE [W3]
  CASE
    [/RA3] :: [GA3] WITH [QA3']
    [/RA4 * RA3] :: [GA4] WITH [QA4']
    [/RA0 * RA4 * RA3] :: [GA0] WITH [QA0']
    [/RA1 * RA0 * RA4 * RA3] :: [GA1] WITH [QA1']
    [/RA2 * RA1 * RA0 * RA4 * RA3] :: [GA2] WITH [QA2']
  ENDCASE

WHILE [W4]
  CASE
    [/RA4] :: [GA4] WITH [QA4']
    [/RA0 * RA4] :: [GA0] WITH [QA0']
    [/RA1 * RA0 * RA4] :: [GA1] WITH [QA1']
    [/RA2 * RA1 * RA0 * RA4] :: [GA2] WITH [QA2']
    [/RA3 * RA2 * RA1 * RA0 * RA4] :: [GA3] WITH [QA3']
  ENDCASE

WHILE [W04]
  CASE
    [/RB0 * RA4 * RA3 * RA2 * RA1 * RA0] :: [GB0] WITH [QB0']
    [/RB1 * RB0 * RA4 * RA3 * RA2 * RA1 * RA0] :: [GB1] WITH [QB1']
    [/RB2 * RB1 * RB0 * RA4 * RA3 * RA2 * RA1 * RA0] :: [GB2] WITH [QB2']
  ENDCASE

WHILE [GA0]
  IF [RA0] THEN [W1] WITH [NOGRANT']

WHILE [GA1]
  IF [RA1] THEN [W2] WITH [NOGRANT']

WHILE [GA2]
  IF [RA2] THEN [W3] WITH [NOGRANT']

WHILE [GA3]
  IF [RA3] THEN [W4] WITH [NOGRANT']

WHILE [GA4]
  IF [RA4] THEN [W0] WITH [NOGRANT']

WHILE [GB0]
  IF [RB0] THEN [GB1] WITH [NOGRANT']

WHILE [GB1]
  IF [RB1] THEN [GB2] WITH [NOGRANT']

WHILE [GB2]
  IF [RB2] THEN [GB0] WITH [NOGRANT']

```

TB00751S

a. Arbiter State Equations (Continued)

Figure 7 (Continued)



# Single Chip Multiprocessor Arbiter

AN7

```

***** ARBITERB *****
***** P I N L I S T *****

```

LABEL	** FNC	**PIN	-----	PIN**	FNC **	LABEL
CLOCK	** CK	** 1-	!	1-28	** +5V	**VCC
RB2	** I	** 2-	!	1-27	** I	**N/C
RB1	** I	** 3-	!	1-26	** I	**N/C
RB0	** I	** 4-	!	1-25	** I	**N/C
RA4	** I	** 5-	!	1-24	** I	**N/C
RA3	** I	** 6-	!	1-23	** I	**N/C
RA2	** I	** 7-	!	1-22	** I	**N/C
RA1	** I	** 8-	!	1-21	** I	**N/C
RA0	** I	** 9-	!	1-20	** I	**N/C
OB2	** 0	** 10-	!	1-19	** PR	**PRESET
OB1	** 0	** 11-	!	1-18	** 0	**OA0
OB0	** 0	** 12-	!	1-17	** 0	**OA1
OA4	** 0	** 13-	!	1-16	** 0	**OA2
GND	** 0V	** 14-	!	1-15	** 0	**OA3

TB00760S

### b. Arbiter Pin List

```

***** ARBITERB *****
@DEVICE TYPE
S25105
@DRAWING
***** MULTI-PROCESSOR BUS ARBITOR
@REVISION
***** ARBITERB REV. 0
@DATE
***** JULY 26, 1985
@SYMBOL
***** ARBITERB
@COMPANY
***** SIGNETICS
@NAME
***** DAVID K. WONG
@DESCRIPTION
@INTERNAL SR FLIP FLOP LABELS
FF0 FF1 FF2 FF3 FF4 FFS
@COMMON PRODUCT TERM
@COMPLEMENT ARRAY
@LOGIC EQUATION

```

TB00770S

### c. Arbiter Boolean Equations

Figure 7 (Continued)



# Single Chip Multiprocessor Arbiter

AN7

## SUMMARY

As can be seen from the circuit diagrams, the FPLS can offer significant advantages over discrete MSI arrays in the design of state machines. The tradeoff in both design alternatives for the Priority Arbiter is shown in Table 2. Clearly, the FPLS approach uses fewer parts, with savings in PC board space and power requirements.

## REFERENCES

1. W.W. Plumber: "Asynchronous Arbiters"; *IEEE Transactions on Computers*, January 1972, pp. 37 - 42.

**Table 3. Design Alternatives for the Priority Arbiter**

PARAMETER	F.P.L.S.	PROM/LATCH
Parts count	1 IC	≈19IC's
PCB space	.84 in <sup>2</sup>	7.92 in <sup>2</sup>
Power	.65W	2.85W
Voltage	+5V	+5V

2. R.C. Pearce, J.A. Field, and W.D. Little: "Asynchronous Arbiter Module"; *IEEE Transactions on Computers*, September 1975, pp. 931 - 933.
3. K. Soe Hojberg: "An Asynchronous Arbiter Resolves Resource Allocation Con-

- flicts on a Random Priority Basis"; *Computer Design*, August 1977, pp. 120 - 123.
4. K. Soe Hojberg: "One-Step Programmable Arbiter for Multiprocessors"; *Computer Design*, April 1978, pp. 154 - 158.

# AN8 Introduction To Signetics Programmable Logic

## Application Note

### Application Specific Products

Author: K. A. H. Noach

### INTRODUCTION

Custom logic is expensive—too expensive if your production run is short. 'Random logic' is cheaper but occupies more sockets and board space. Signetics Programmable Logic bridges the gap. Using PLD, you can configure an off-the-shelf chip to perform just the logic functions you need. Design and development times are much shorter, and risk much lower than for custom logic. Connections are fewer than for random logic, and, for all but the simplest functions, propagation delay is usually shorter. Yet another advantage that PLD has over custom logic is that it allows you to redesign the functions without redesigning the chip—giving you an invaluable margin not only for cut-and-try during system development, but also for later revision of system design. You're not tied down by the need to recover capital invested in a custom chip.

A PLD chip is an array of logic elements—gates, inverters, and flip-flops, for instance. In the virgin state, everything is connected to everything else by nichrome fuses, and although the chip has the capacity to perform an extensive variety of logic functions, it doesn't have the ability to. What gives it that is programming: selectively blowing undesired fuses so that those that remain provide the interconnections necessary for the required functions.

Signetics Series 20 PLD, named for the number of pins, supplements the well-known Series 28. The package is smaller—little more than a third the size, in fact—but the improved architecture, with user-programmable shared I/O, compensates for the fewer pins. The series comprises the following members, in order of increasing complexity:

- PLS151—field-programmable gate array
- PLS153—field-programmable logic array

- PLS155—field-programmable logic sequencer
- PLS157—field-programmable logic sequencer
- PLS159—field-programmable logic sequencer

Entry to all the devices is via a product matrix, an array of input and shared I/O lines fuse-connected to the multiple inputs of an array of AND gates (see Figures 1, 2 and 5). To exploit the capacity of any device, it is important to make the most economical use of the AND gates it has available. Application of de Morgan's theorem can help in this. For example, inputs for the function

$$F = A + B + C + D$$

would occupy four of the AND gates of the product matrix. However, the same function rewritten as

$$\bar{F} = \bar{A} \bar{B} \bar{C} \bar{D}$$

would occupy only one. Moreover, the second function could be done on the simplest of the Series 20 devices (and leave eleven gates over for other functions), whereas the first could not. The fact that all inputs of the Series 20 devices, including the shared ones, incorporate double buffers that make the true and complement forms of all input variables equally accessible, greatly facilitates the use of de Morgan's theorem for logic minimization.

To convert the minimized logic equations to the pattern of fuses to be blown, you can use either a programming sheet (see e.g. Table 1) or Boolean equation program-entry software that lets you enter the equations via the keyboard of a terminal. The direct programmability of logic equations makes system design with PLD simple and sure. Functional changes can be made by replacing one PLD chip by another differently programmed. In many cases you can even remove the original one, reprogram it on the spot, and re-insert it. Programming machines qualified for the Se-

ries 20 are at present available from DATA I/O, KONTRON, and STAG.

### FPGA PLS151

The field-programmable gate array is the simplest of the Series 20 PLD devices; Figure 1 shows the functional diagram. The array can accept up to 18 inputs. There are six dedicated input pins (A) and twelve (A') that can be programmed as inputs, outputs, or bidirectional I/O. All input variables, whether on dedicated or programmed input pins, are available in both true and complement form in the product matrix (B), and both forms are buffered: either form can drive all 12 product lines if required. In the virgin state, all the input variables and their complements are connected to all the product lines via a diode and a fuse (C), and the product matrix is effectively inoperative. To enable it to generate the required functions, unrequired connections between individual input lines and product lines are severed by blowing the connecting fuses.

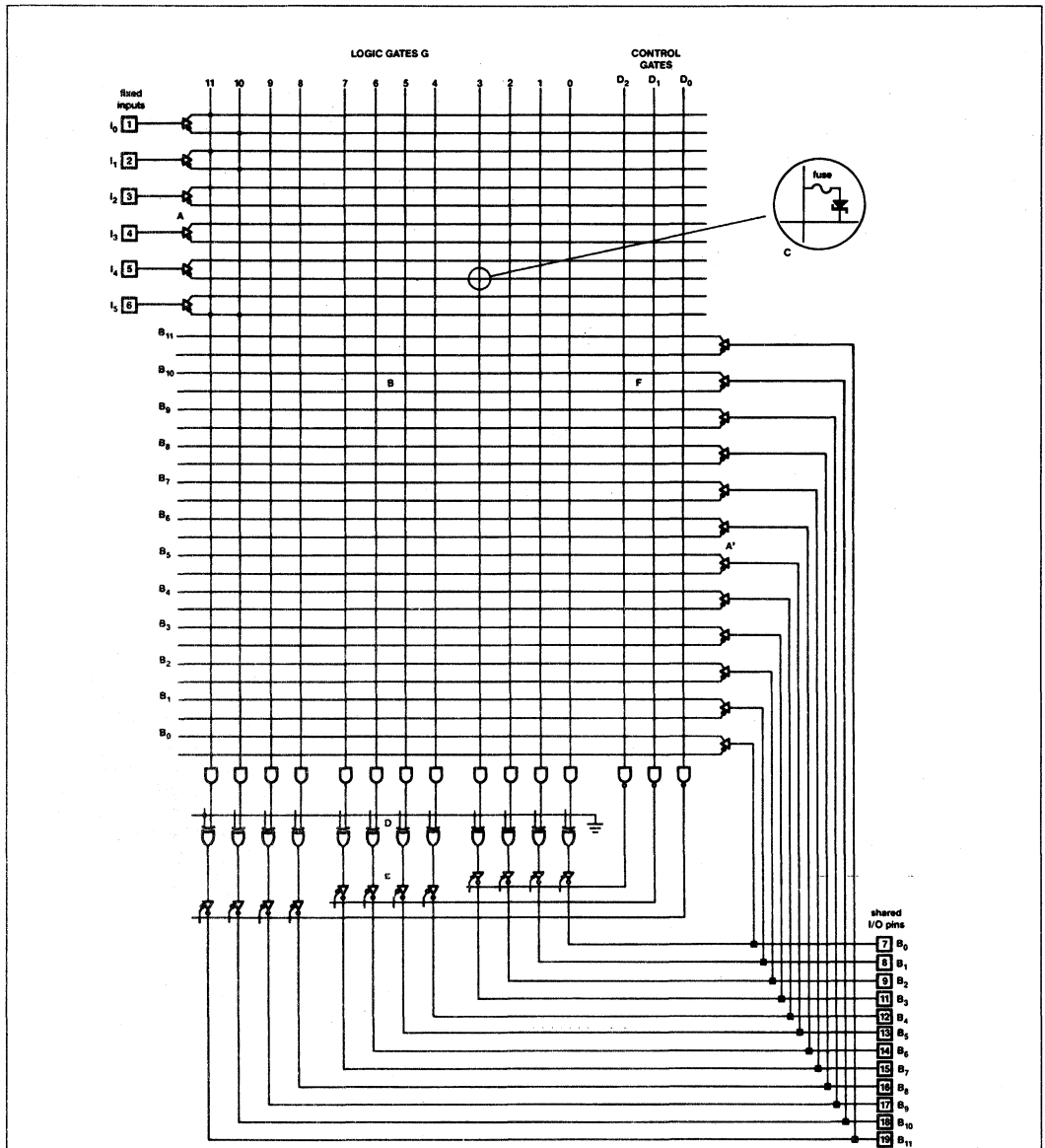
At the output of the product matrix are 12 NAND gates, each with 36 inputs to accommodate the 18 possible input variables and their complements. Each of the product terms is normally active-Low, but a unique feature of Signetics PLD is that any or all of them can be independently programmed active-High. This is done by means of an array of exclusive-OR gates (D) at the NAND-gate outputs; when the fuse that grounds the second input of each OR gate is blown, the output of that gate is inverted.

The product matrix and exclusive OR-gate connections shown in Figure 1 illustrate the flexibility conferred by having buffered complements of all input variables internally available, together with independently programmable output polarities. Output B<sub>11</sub>, shown with its exclusive OR-gate fuse intact, is programmed

$$\bar{B}_{11} = I_0 I_1 \bar{I}_5$$

# Introduction To Signetics Programmable Logic

AN8



**NOTE:**

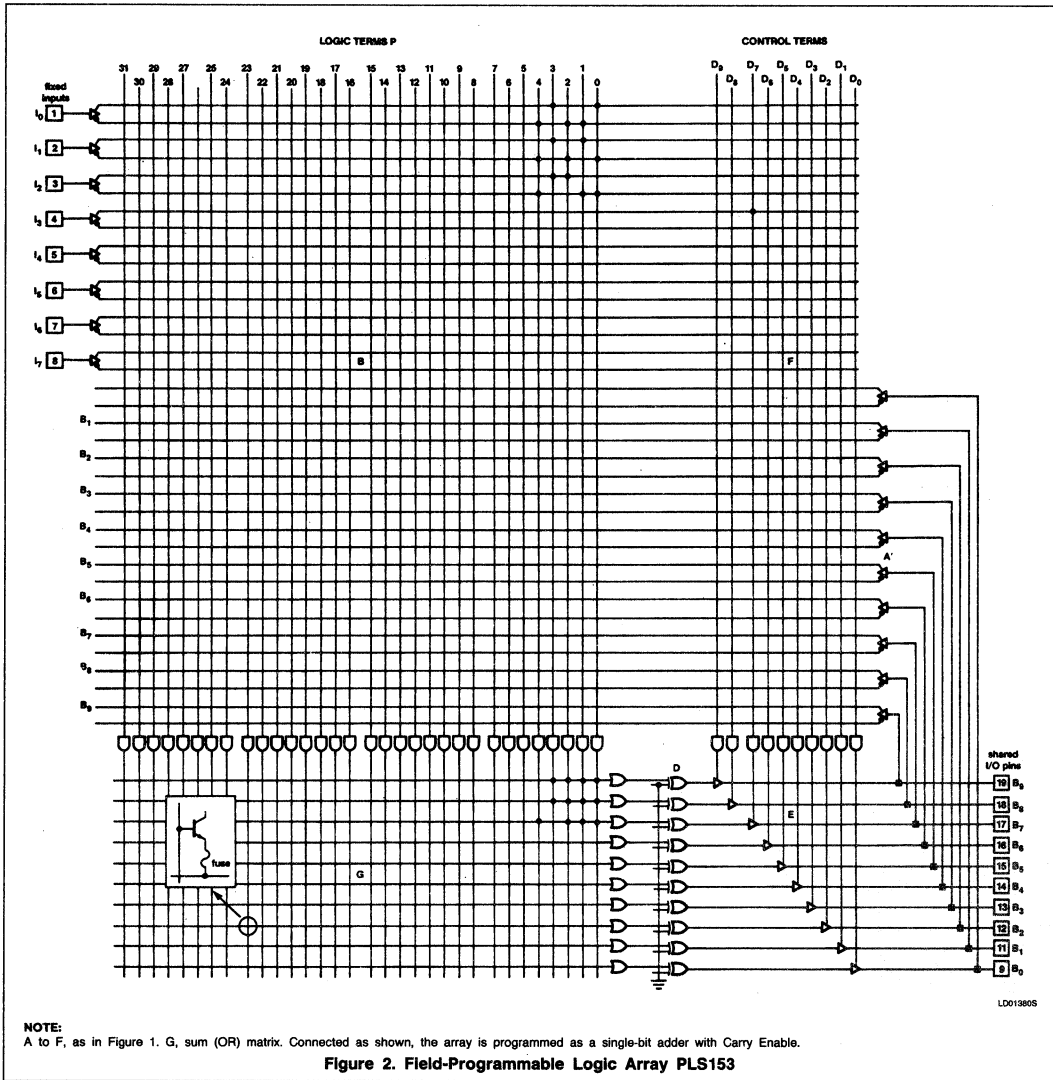
A, dedicated inputs; A', programmable I/O. B, product (NAND) matrix with fused connections C, each of the vertical lines in the matrix represents 36 inputs to the terminating NAND gated. D, exclusive-OR array with inputs grounded via fuses for polarity control. E, programmable Tri-state output buffers. F, fuse-programmable control matrix. Square dots (■) represent permanent connections; round dots (●) intact fuse connections. Connected as shown, the array is programmed for the functions  $B_{11} = I_0 I_1 I_5$  and  $B_{10} = I_0 I_1 I_5$ .

**Figure 1. Field-Programmable Gate Array PLS151A**

LD013415

# Introduction To Signetics Programmable Logic

AN8



At the same time, and without using any additional inputs, output B<sub>10</sub> (fuse blown) is programmed

$$B_{10} = I_0 I_1 \bar{I}_5$$

Each of the exclusive-OR gates drives a three-state output buffer. In the virgin state all the buffers (E) are disabled and therefore in the high-impedance state. The function of the programmable I/O pins (A') is then determined by the I/O control matrix (F). The three AND gates at the control-matrix output are

Active-High, and when one of them is in the High state, the four output buffers it controls are enabled; the corresponding I/O pins then act as outputs. Conversely, when a control-matrix AND-gate output is Low and the control fuse for the corresponding Tri-state buffer is intact, the pins controlled by that gate act as inputs. Thus, these pins can be programmed in groups of up to four to act as inputs or outputs according to the state of selected input variables. If required, any of the programmable I/O pins can be made a

dedicated output by blowing the control fuse of the output buffer associated with it.

The speed of the FPGA compares favorably with TTL, although its propagation delay is longer than the individual gate delay of TTL. When the number of inputs required is large, however, the FPGA more than makes up for this. When more than eight inputs are required, for example, the FPGA has a distinct advantage. Then, the overall propagation delay

# Introduction To Signetics Programmable Logic

# AN8

lay of TTL often amounts to two or three gate delays, but that of the FPGA to only one.

## FPLA PLS153

### Architecture

With two levels of logic embodied in a product matrix terminating in 32 AND gates coupled to a ten-output OR matrix (Figure 2), the FPLA is a step up in complexity from the FPGA. Again, there is provision for 18 input variables, internally complemented and buffered, but here divided between eight dedicated input pins and ten individually programmable I/O pins. As before, exclusive-OR gates grounded by fuses provide output polarity control, and any of the programmable I/O pins can be made a dedicated output by blowing the control fuse of the output buffer associated with it.

### Programming

When the required functions have been defined, corresponding programming instructions are entered in a programming table, the layout of which reflects the FPLA architecture. (A Signetics computer program named AMAZE, which accepts Boolean equations as input and generates an FPLA programming table as output, is also available.) The programming machine blows the FPLA fuses in the pattern prescribed by the table.

As an illustration of FPLA programming, consider a full adder. Figure 3 shows a TTL version (74LS80) and the corresponding logic equations. Note that the feedback of  $\bar{C}_{n+1}$  introduces a second propagation delay. In the FPLA this is eliminated by redefining  $\Sigma$  in terms of A, B, and  $C_n$ , as shown in Figure 4, and using the right side of the equation for  $\bar{C}_{n+1}$  instead of the term itself. At first glance this would appear to require a minimum of three product terms for  $\bar{C}_{n+1}$  plus four for  $\Sigma$ , or a total of seven. The Karnaugh maps, however, show considerable overlap between the two functions: the map for  $\bar{C}_{n+1}$  differs from that for  $\Sigma$  only by having A B  $C_n$  instead of  $\bar{A} \bar{B} \bar{C}_n$ . Rewriting the equation for  $\bar{C}_{n+1}$  to introduce  $\bar{A} \bar{B} \bar{C}_n$  and eliminate A B  $C_n$ ,

$$\bar{C}_{n+1} = A \bar{B} \bar{C}_n + \bar{A} B \bar{C}_n + \bar{A} \bar{B} C_n + \bar{A} \bar{B} \bar{C}_n$$

increases the number of product terms by one, but now  $\bar{C}_{n+1}$  and  $\Sigma$  have three terms in common. Therefore, since the FPLA allows multiple use of product terms, it is sufficient to program each of the common terms only once; thus, the original seven product terms are effectively reduced to five.

To fill in the programming table (Table 1), first allocate inputs and outputs.

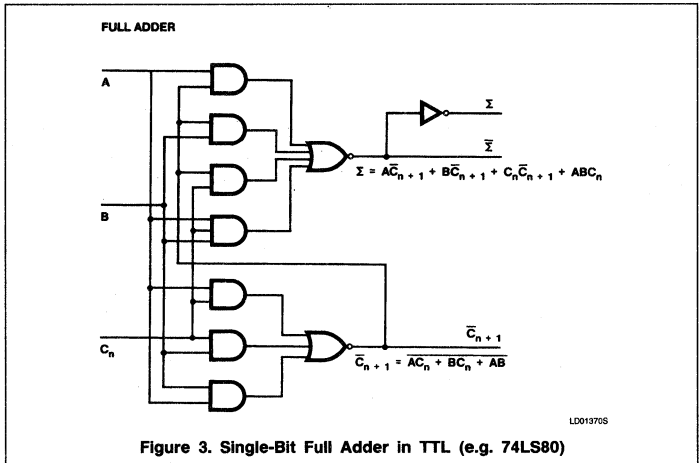


Figure 3. Single-Bit Full Adder in TTL (e.g. 74LS80)

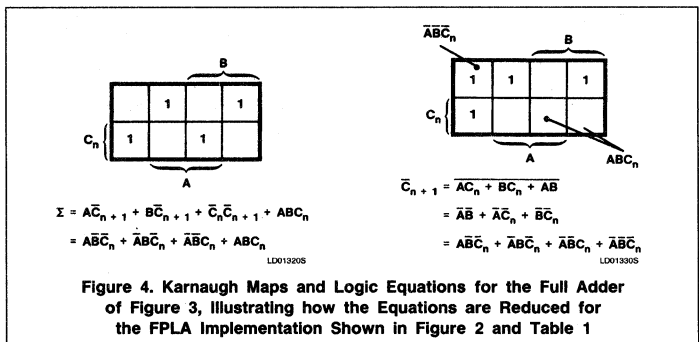


Figure 4. Karnaugh Maps and Logic Equations for the Full Adder of Figure 3, illustrating how the Equations are Reduced for the FPLA Implementation Shown in Figure 2 and Table 1

Inputs: A = I<sub>0</sub>      Outputs:  $\bar{C}_{n+1}$  = B<sub>7</sub>  
 B = I<sub>1</sub>              Σ = B<sub>8</sub>  
 C<sub>n</sub> = I<sub>2</sub>              Σ = B<sub>9</sub>

Next, enter the product terms of  $\Sigma$  in the product-matrix (AND) part of the table, using H to indicate a true input and L a false one.

- Term 0 is A  $\bar{B} \bar{C}_n$ : mark H, L, L in columns I<sub>0</sub>, I<sub>1</sub>, I<sub>2</sub> of row 0
- Term 1 is  $\bar{A} B \bar{C}_n$ : mark L, H, L in columns I<sub>0</sub>, I<sub>1</sub>, I<sub>2</sub> of row 1
- Term 2 is  $\bar{A} \bar{B} C_n$ : mark L, L, H in columns I<sub>0</sub>, I<sub>1</sub>, I<sub>2</sub> of row 2
- Term 3 is A B  $C_n$ : mark H, H, H in columns I<sub>0</sub>, I<sub>1</sub>, I<sub>2</sub> of row 3.

Fill the rest of rows 0, 1, 2, and 3 with dashes to indicate that all other inputs are to be disconnected from Terms 0, 1, 2, and 3 (fuses blown).

The product terms of  $\Sigma$  must be added to form the sum-of-products required at output

B<sub>9</sub>. Indicate the required addition by putting an A (for Attached, i.e. fuse unblown) in the Term 0, 1, 2, and 3 spaces of column B(O)<sub>9</sub>; Term 4 is not required for  $\Sigma$ , so put a dot in the Term 4 space to indicate that it is to be disconnected (fuse blown). To indicate that the output is to be Active-High, put an H in the polarity square above the B(O)<sub>9</sub> column. Finally, fill row D<sub>9</sub> with dashes to indicate that all fuses on line D<sub>9</sub> of the control matrix are to be blown and B<sub>9</sub> is to be a dedicated output. This completes the programming of  $\Sigma$ .

The  $\bar{\Sigma}$  output on B<sub>8</sub> is programmed in just the same way, except that the polarity square above the B(O)<sub>8</sub> column is marked L to indicate Active-Low. (Note that in the FPLA, the  $\Sigma$  and  $\bar{\Sigma}$  outputs change simultaneously, because all output signals traverse the exclusive-OR array (D), whether they are Active-High or Active-Low. In the TTL full adder shown in Figure 3, the output inverter delays the change of  $\Sigma$  with respect to  $\bar{\Sigma}$ .)

# Introduction To Signetics Programmable Logic

AN8

Table 1. FPLA Programming Table Filled in for the Full Adder of Figure 2

TERM	AND																POLARITY											
	I								B(i)								OR											
	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0
																	B(O)											
0	-	-	-	-	-	-	L	L	H	-	-	-	-	-	-	-	-	A	A	A								
1	-	-	-	-	-	-	L	H	L	-	-	-	-	-	-	-	-	A	A	A								
2	-	-	-	-	-	-	H	L	L	-	-	-	-	-	-	-	-	A	A	A								
3	-	-	-	-	-	-	H	H	H	-	-	-	-	-	-	-	-	A	A	•								
4	-	-	-	-	-	-	L	L	L	-	-	-	-	-	-	-	-	•	•	A								
5																												
6																												
7																												
8																												
9																												
10																												
11																												
12																												
13																												
14																												
15																												
16																												
17																												
18																												
19																												
20																												
21																												
22																												
23																												
24																												
25																												
26																												
27																												
28																												
29																												
30																												
31																												
D9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-											
D8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-											
D7	-	-	-	-	H	-	-	-	-	-	-	-	-	-	-	-	-											
D6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-											
D5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-											
D4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-											
D3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-											
D2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-											
D1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-											
D0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-											
PIN NO.	8	7	6	5	4	3	2	1	19	18	17	16	15	14	13	12	11	9										
VARIABLE NAME					ENABLE	C <sub>n</sub>	B	A	Σ	Σ	C <sub>n+1</sub>																	

T801480S



## Introduction To Signetics Programmable Logic

AN8

The output  $C_{n+1}$  on  $B_7$  contains three of the same terms as  $\Sigma$ , plus the term  $\bar{A} \bar{B} \bar{C}_n$ . Only this last term needs to be additionally programmed in the product matrix: mark L, L, L in columns  $I_0, I_1, I_2$  of the Term 4 row. Indicate the addition

$$A \bar{B} \bar{C}_n + \bar{A} B \bar{C}_n + \bar{A} \bar{B} C_n + \bar{A} \bar{B} \bar{C}_n$$

by putting an A in rows 0, 1, 2, and 4 of column  $B(O)_7$ , and show that Term 3 ( $A B C_n$ ) is not required by putting a dot in the Term 3 row to indicate disconnection (fuse blown). Put an L in the  $B(O)_7$  polarity square to indicate Active-Low.

Identifying  $B_7$  as a dedicated output by indicating that all the fuses to control term  $D_7$  are to be blown, would now complete the programming of the full adder. However, a useful supplementary feature would be a Carry Enable function to keep the  $B_7$  output buffer in the high-impedance state except when the enable input  $I_3$  is true. The output buffer is enabled when both the fuses of a control term are blown, or when one is blown and the term that controls the output buffer is true. Thus, a Carry Enable can be provided via the  $I_3$  input by leaving intact the fuse for Active-High operation of the enable signal to control term  $D_7$ . To indicate this, put an H in the  $I_3$  column of row  $D_7$  and fill the rest of the row with dashes.

The full adder with output Carry Enable uses only four of the eight dedicated inputs, three of the ten programmable I/O pins, and five of the 32 AND gates. The remaining capacity can be used for programming other functions which may, if required, also make use of AND-gate outputs already programmed for the full adder.

All fuses not indicated as blown in the programming table are normally left intact to preserve capacity for later program revisions or the addition of supplementary functions. If it is essential to minimize propagation delay, however, the finalized program should include instructions for blowing all unused fuses to minimize load capacitance.

### FPLS PLS155 - PLS157 - PLS159 Architecture

The FPLS (Figure 5) is the most complex of the Series 20 PLD devices. Like the FPLA, it has a 32-term product matrix followed by an OR matrix. In the FPLS, however, the OR matrix is larger and comprises three distinct parts, with architecture differing in detail from type to type. In the PLS155, for instance, the first part consists of eight 32-input gates coupled, like those of the FPLA, to an output-polarity-controlling exclusive-OR array. The

second consists of twelve additional gates which control four flip-flops. These are what give the FPLS its sequential character, enabling it to dictate its next state as a function of its present state. The third part is the deceptively simple Complement Array (I in Figure 5): a single OR gate with its output inverted and fed back into the product matrix. This enables a chosen sum-of-products to become a common factor of any or all the product terms and makes it possible to work factored sum-of-products equations. It is also useful for handshaking control when interfacing with a processor and for altering the sequence of a state machine without resorting to a large number of product terms.

PLS155 has four dedicated inputs and eight programmable I/O pins that can be allocated in the same way as in the FPLA. It also has four shared I/O pins (L) whereby the flip-flops can be interfaced with a bidirectional data bus. Two product terms,  $L_A$  and  $L_B$  in the control matrix F, control the loading of the flip-flops, in pairs, synchronized with the clock.

Figure 6 shows the architecture of the flip-flop circuitry in the PLS155. The flip-flops are positive-edge-triggered and can be dynamically changed to J-K, T, or D types according to the requirements of the function being performed; this considerably lessens the demands on the logic. The Tri-state inverter between the J and K inputs governs the mode of operation, under the control of the product term F:

- When the inverter is in the high-impedance state, the flip-flop is a J-K type, or a T type when  $J = K$ .
- When the inverter is active,  $K = \bar{J}$  and the flip-flop is a D type; the K input must then be disconnected from the OR matrix.

All the product terms from the product matrix ( $T_0$  to  $T_{31}$  in Figure 5) are fuse-connected to the J and K input OR gates. If both fuses of any one product term are left intact,  $J = K$  and the flip-flop is a T type.

The flip-flops of the PLS155 have asynchronous Preset and Reset controlled by terms in the OR matrix that take priority over the clock. Their three-state output buffers can be controlled from the enable pin OE or permanently enabled or disabled by blowing fuses or leaving them intact in the enable array (K in Figure 5).

The PLS157 and PLS159 sequencers have, respectively, six and eight flip-flops. The architecture differs in detail but is similar in principle to that of PLS155.

### Programming

The FPLS is programmed in much the same way as the FPLA, using a table to instruct the machine that blows the undesired fuses. It is not necessary to work with a circuit diagram; in fact, it is even undesirable to do so, since applying the necessary logic reduction techniques would in most cases make the diagram difficult to read and more a hindrance than a help. An example of how to program the FPLS as a universal counter/shift-register is given in the Appendix.

### DEVELOPMENT AND PRODUCTION ECONOMY WITH PLD

Underlying the design philosophy of the Signetics Series 20 PLD is the concept of programmable arrays whose architecture emulates logic equation formats rather than mere aggregations of gates. The unique combination of features which support this philosophy includes:

- double-buffered true and complement inputs
- programmable-polarity outputs
- programmable I/O for internal feedback and maximum freedom in allocating inputs and outputs
- truth-table programming format

These features are common to all the PLD devices. In the field-programmable logic sequencers they are further supported by:

- flip-flops with dynamically alterable operating modes
- a complement array for simplified handshaking control

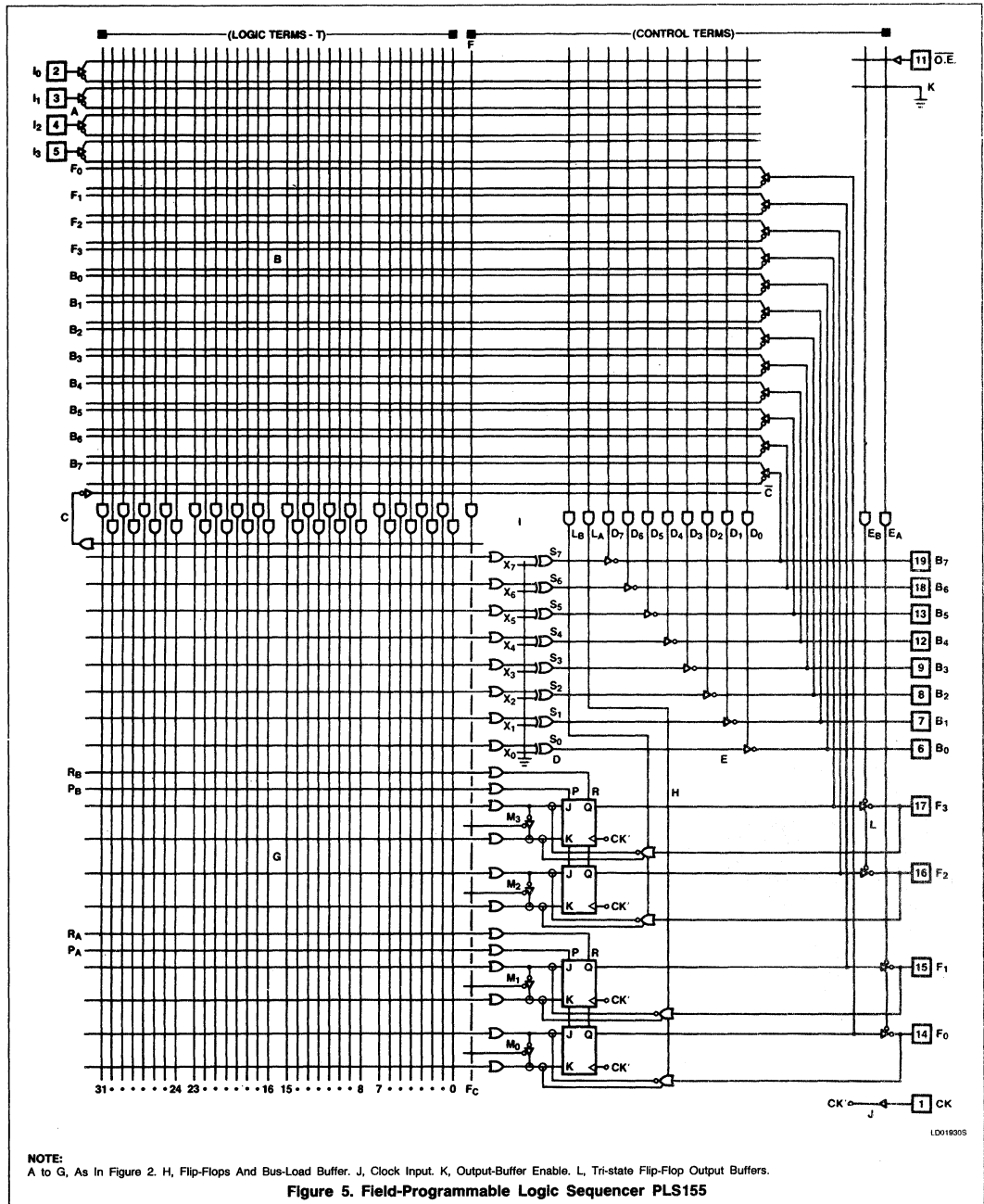
From the development engineer's point of view an important advantage of PLD is that it eliminates breadboarding. Once the functions required in terms of minimized logic equations are worked out, a PLD can be programmed accordingly. Once programmed, it will perform those functions.

Loading the instructions into the programming machine usually takes no more than a couple of hours; after that, the machine can program the devices at a rate of 100 an hour. Moreover, since any PLD can be programmed in many different ways, PLD has considerable potential for simplifying purchasing and stock control. One type of device can be programmed to perform a diversity of tasks for which it would otherwise be necessary to purchase and stock many different devices.

Series 20 PLD is second-sourced by Harris Semiconductor.

# Introduction To Signetics Programmable Logic

AN8



# Introduction To Signetics Programmable Logic

AN8

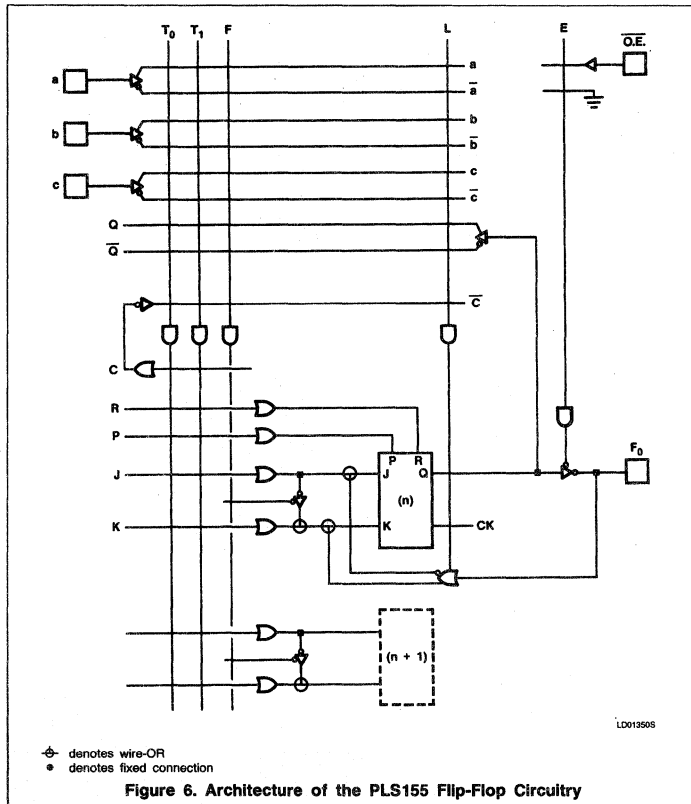


Figure 6. Architecture of the PLS155 Flip-Flop Circuitry

## APPENDIX

### Programming an FPLS as a Counter/Shift-Register

Objective: to program a PLS155 FPLS as a count-up, count-down, shift-right, shift-left machine governed by three control terms - COUNT/SHIFT, RIGHT/UP, LEFT/DOWN. Direct implementation would result in a machine with 64 state transitions (see Table A-1), which is beyond the scope of the PLS155 or even the 28-pin PLS105. Logic reduction is therefore necessary.

As there are only four feedback variables (D, C, B, A), you can do the reduction by hand, one mode at a time; the control terms need not be included till the summary equations are written. Using the transition mapping method suggested here, you can examine the excitation equations for all types of flip-flops

(R-S, J-K, D, T) and choose those types that will perform the required functions using the fewest product terms. Table A-2 summarizes the rules for flip-flop implementation using transition maps; the transition symbols used in the table mean:

PRESENT STATE	NEXT STATE	TRANSITION SYMBOL
0	0	0
0	1	$\alpha$
1	0	$\beta$
1	1	1

Using these symbols, construct Table A-3 from Table A-1 to enable you to examine the excitation equations for all types of flip-flops. Proceeding one mode at a time, transfer the state conditions from Table A-3 to Karnaugh maps, as in Figure A-1. Following the rules in

Table A-2, derive the excitation equations for the different types of flip-flops (the examples shown in Figure A-1 omit the T type because it is the same as the J-K type when  $J = K$ ). In deciding which types of flip-flop to use, remember that logic minimization with PLD is different from logic minimization with 'random logic': with random logic you seek to reduce the number of standard packages required; with PLD you seek to reduce the number of product terms.

From Figure A-1 it is evident that you should choose J-K or T flip-flops for the counter mode and D flip-flops for the shift mode, for you then require only one product term per flip-flop per mode. Table A-4 summarizes the number of product terms per mode the various types of flip-flops would require.

Table A-5 shows the completed programming table for the counter/shift-register. The programming of Terms 0 to 15 reflects the flip-flop excitation equations and illustrates the value of being able to switch the flip-flops dynamically from one type of operation to another. Terms 16, 17 and 18, respectively, provide for INITIALIZE, asynchronous RESET, and STOP functions.

The programming of the two additional inputs HALT and BUSY illustrates the value of the complementary, which is made active when HALT and BUSY are Low (A in the Complement square of Term 18) and propagated into all the other terms (dot in the Complement squares of Terms 0 to 17). This means that unless the HALT and BUSY inputs are High, none of the product terms will be true and the state of the machine will not change. If the Complement Array were not used, twice the number of product terms would be required, even if one of the additional inputs were omitted.

As it is, the design uses only 19 of the 32 product terms available, so there is ample capacity for extending its capabilities. For example, the shift-left function can be augmented by a binary multiplication capability, using a D type flip-flop to make it shift one, two, or three places according to the state of two extra inputs, X and Y. Table A-6 shows the revised programming table. The binary multiplication function occupies nine additional product terms.

## ACKNOWLEDGEMENT

*Electronic Components and Applications*; Vol. 4, No. 2, February 1982. Reprinted with the permission of PHILIPS.

# Introduction To Signetics Programmable Logic

AN8

**Table A-1. Present-State/Next-State Table for Counter/Shift-Register**

STATE NO.	PRESENT STATE				NEXT STATE															
					Count Down				Count Up				Shift Left				Shift Right			
0	D	C	B	A	D	C	B	A	D	C	B	A	D	C	B	A				
1	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0				
2	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0				
3	0	0	1	0	0	0	0	1	0	0	1	0	0	1	0	0				
4	0	0	1	1	0	0	1	0	0	1	0	0	0	1	1	0				
5	0	1	0	0	0	0	1	1	0	1	0	1	1	0	0	0				
6	0	1	0	1	0	1	0	0	0	1	1	0	1	0	1	0				
7	0	1	1	0	0	1	0	0	0	1	1	0	1	1	0	0				
8	0	1	1	1	0	1	1	0	1	1	0	0	1	1	1	0				
9	1	0	0	0	0	1	1	1	1	0	0	0	0	0	1	0				
10	1	0	0	1	1	0	0	0	1	0	1	0	0	1	1	0				
11	1	0	1	0	1	0	1	0	1	1	0	0	0	1	1	1				
12	1	0	1	1	1	0	1	0	1	1	1	0	1	1	1	0				
13	1	1	0	0	1	1	0	0	1	1	1	0	1	0	1	1				
14	1	1	0	1	1	1	0	0	1	1	1	1	1	1	0	1				
15	1	1	1	0	1	1	1	0	0	0	0	0	1	1	1	1				
<b>CONTROL TERMS</b>																				
COUNT/SHIFT					1				1				0				0			
RIGHT/UP					0				1				0				1			
LEFT/DOWN					1				0				1				0			

**Table A-2. Rules for Flip-Flop Implementation Using Transition Maps**

FLIP-FLOP TYPE	INPUT	MUST INCLUDE	MUST EXCLUDE	REDUNDANT
R-S	S	$a$	$\beta, 0$	$1, x$
	R	$\beta$	$a, 1$	$0, x$
D	D	$a, 1$	$\beta, 0$	$x$
T	T	$a, \beta$	$0, 1$	$x$
J-K	J	$a$	$0$	$1, \beta, x$
	K	$\beta$	$1$	$0, a, x$







# AN10

## 4-Bit Binary-To-7 Segment Decoder

### Application Note

### Application Specific Products

### INTRODUCTION

Using the PLS153 or PLS153A, a 4-bit binary-to-7 segment decoder may be easily constructed. This paper will cover Hex-to-7 segment decoder, BCD-to-7 segment decoder, decoder with latches, and decoder with multiplexer. The architecture of PLS153/153A is basically an AND-OR matrix. The large number of inputs of each AND term is ideal for decoder applications, and the ability to drive the entire OR array makes possible the efficient use of AND terms such that, together with the programmable output polarity, a minimum number of AND terms is needed to implement the decoder function. Another important feature of the PLS153/153A architecture is the 10 bidirectional I/O pins. The bidirectionality of these pins provides internal feedback to the AND matrix, which eliminates the need for external wiring and, more importantly, it saves I/O pin requirements. All I/O pins of the PLS153/153A have tri-state output buffers, each may be enabled or disabled by its own controlling AND terms. In applications where high current drive is necessary but low power consumption is still desirable, the AND term controlled tri-state buffers are ideal for multiplexing. The LED display may be driven at a higher current and lower duty cycle to get better brightness/current efficiency and reduce the overall power dissipation.

### DESCRIPTION

#### Hex-To-7 Segment Decoder

A 7 segment display (Figure 1) may be used to represent a hexadecimal number as shown in Table 1. The format serves merely as an example. Some designers may prefer other configurations, particularly for letters B and D. Before implementing the truth table with programmable logic devices, one has to decide whether the outputs should be Active-High or Active-Low. If a moderate drive current of 15mA or less is needed, and if a common anode LED

display is used, then an Active-Low output configuration is sufficient. On the other hand, if only a common cathode LED display is available and the current drive requirement is not very critical, an Active-High output configuration may be used. The change of output polarity may be easily affected by putting in the POLARITY section 'H' for Active-High and 'L' for Active-Low. Since the output polarity is programmable, one may choose to describe the logic in terms of outputs equal to '1's or outputs equal to '0's.

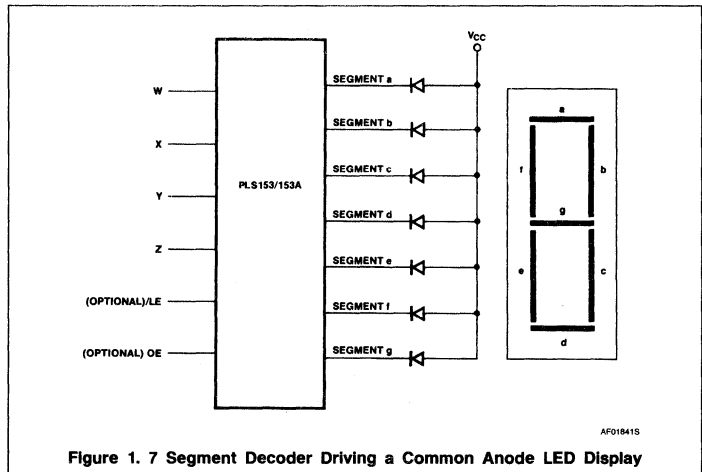


Figure 1. 7 Segment Decoder Driving a Common Anode LED Display

### A>\*\*\*\*\* P I N L I S T \*\*\*\*\*

LABEL	** FNC	**PIN	-----	PIN**	FNC **	LABEL
W	** I	** 1-	----	!-20	** +5V	**VCC
X	** I	** 2-	----	!-19	** B	**N/C
Y	** I	** 3-	----	!-18	** B	**N/C
Z	** I	** 4-	8	!-17	** B	**/SEG_A
N/C	** I	** 5-	2	!-16	** B	**/SEG_B
N/C	** I	** 6-	5	!-15	** B	**/SEG_C
N/C	** I	** 7-	1	!-14	** B	**/SEG_D
/OE	** I	** 8-	5	!-13	** B	**/SEG_E
N/C	** B	** 9-	3	!-12	** B	**/SEG_F
GND	** OV	** 10-	!	!-11	** B	**/SEG_G

TB00981S

Figure 2. Pin List of Binary-to-7 Segment Decoder

### NOTE:

PLSXXX and 82SXXX are interchangeable part numbers.



# 4-Bit Binary-To-7 Segment Decoder

AN10

Table 1. Truth Table of Hex-to-7-Segment Decoder

The diagram illustrates the 7-segment outputs for hexadecimal digits 0 through F. Each digit is shown with its corresponding 4-bit binary input (ZYXW) and 7-bit segment output (abcdefg). The segments are labeled a through g, corresponding to the standard 7-segment display layout.

ZYXW	a	b	c	d	e	f	g
0000	1	1	1	1	1	0	0
0001	0	1	1	0	0	0	0
0010	1	1	0	1	1	0	1
0011	1	1	1	1	0	0	1
0100	0	1	1	0	0	1	1
0101	1	0	1	1	0	1	1
0110	1	0	1	1	1	1	1
0111	1	1	1	0	0	0	0
1000	1	1	1	1	1	1	1
1001	1	1	1	1	0	1	1
1010	1	1	1	0	1	1	1
1011	0	0	1	1	1	1	1
1100	1	0	0	1	1	1	0
1101	0	1	1	1	1	0	1
1110	1	0	0	1	1	1	1
1111	1	0	0	0	1	1	1

TB009803

# 4-Bit Binary-To-7 Segment Decoder

AN10

Table 2. H/L Programming Table for Binary-to-7-Segment Decoder

82S153/153A PROGRAMMING TABLE      DRAWING ID: B TO 7 SEG DECODER      REV. —  
 DESIGNER: D. K. WONG      DATE: 12/15/84

	AND														OR														COMMENTS	
	A							B(1)							B(0)							INPUT	OUTPUT							
	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7			6	5	4	3	2	1	0
0	—	—	—	—	L	L	L	L	—	—	—	—	—	—	—	—	—	—	••	A	A	A	A	A	A	••	0	0		
1	—	—	—	—	L	L	L	H	—	—	—	—	—	—	—	—	—	—	••	A	A	••	••	••	••	••	1	1		
2	—	—	—	—	L	L	H	L	—	—	—	—	—	—	—	—	—	—	••	A	A	••	A	A	••	••	2	2		
3	—	—	—	—	L	L	H	H	—	—	—	—	—	—	—	—	—	—	••	A	A	A	••	••	A	••	3	3		
4	—	—	—	—	L	H	L	L	—	—	—	—	—	—	—	—	—	—	••	A	A	••	••	A	A	••	4	4		
5	—	—	—	—	L	H	L	H	—	—	—	—	—	—	—	—	—	—	••	A	••	A	••	A	A	••	5	5		
6	—	—	—	—	L	H	H	L	—	—	—	—	—	—	—	—	—	—	••	A	••	A	A	A	••	••	6	6		
7	—	—	—	—	L	H	H	H	—	—	—	—	—	—	—	—	—	—	••	A	A	••	••	••	••	••	7	7		
8	—	—	—	—	H	L	L	L	—	—	—	—	—	—	—	—	—	—	••	A	A	A	A	A	A	••	8	8		
9	—	—	—	—	H	L	L	H	—	—	—	—	—	—	—	—	—	—	••	A	A	A	••	A	A	••	9	9		
10	—	—	—	—	H	L	H	L	—	—	—	—	—	—	—	—	—	—	••	A	A	••	A	A	••	A	••	10	A	
11	—	—	—	—	H	L	H	H	—	—	—	—	—	—	—	—	—	—	••	A	A	A	A	••	••	••	11	b		
12	—	—	—	—	H	H	L	L	—	—	—	—	—	—	—	—	—	—	••	A	••	A	A	A	A	••	12	c		
13	—	—	—	—	H	H	L	H	—	—	—	—	—	—	—	—	—	—	••	A	••	A	A	••	A	••	13	d		
14	—	—	—	—	H	H	H	L	—	—	—	—	—	—	—	—	—	—	••	A	••	A	A	••	A	••	14	E		
15	—	—	—	—	H	H	H	H	—	—	—	—	—	—	—	—	—	—	••	A	••	••	A	A	••	••	15	F		
16																														
17																														
18																														
19																														
20																														
21																														
22																														
23																														
24																														
25																														
26																														
27																														
28																														
29																														
30																														
31																														
D9																														
D8																														
D7	L	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—													
D6	L	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—													
D5	L	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—													
D4	L	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—													
D3	L	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—													
D2	L	L	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—													
D1	L	L	L	—	—	—	—	—	—	—	—	—	—	—	—	—	—													
D0	L	L	L	L	—	—	—	—	—	—	—	—	—	—	—	—	—													
N																														
A																														
M																														
E																														

ENABLE B7 FOR OUTPUT (SEG A)  
 " B6 " " " B  
 " B5 " " " C  
 " B4 " " " D  
 " B3 " " " E  
 " B2 " " " F  
 " B1 " " " G

A B C D E F G

# 4-Bit Binary-To-7 Segment Decoder

AN10

Table 3. H/L Programming Table for Binary-to-7 Segment Decoder

82S153/153A PROGRAMMING TABLE DRAWING ID: B-7 SEG. DECODER REV. A  
 DESIGNER: D. C. WONG DATE: 12/15/84

A	AND														OR														COMMENTS															
	B(1)							B(0)							B(1)							B(0)																						
	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0																
0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0
1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1
2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2
3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3
4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	4
5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	5
6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	6
7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	7
8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	8 (*)
9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	9
10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	10
11	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	11
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	12
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	13
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	14
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	15
16																																												
17																																												
18																																												
19																																												
20																																												
21																																												
22																																												
23																																												
24																																												
25																																												
26																																												
27																																												
28																																												
29																																												
30																																												
31																																												
D9																																												
D8																																												
D7	H																																											
D6	H																																											
D5	H																																											
D4	H																																											
D3	H																																											
D2	H																																											
D1	H																																											
D0	H																																											
N																																												
A																																												
M																																												
E																																												

(\*) SINCE THIS SIGNAL IS "INACTIVE", IT MAY BE DELETED. THEREFORE, THE TOTAL # OF TERMS REQUIRED IS 15.

TB010115

# 4-Bit Binary-To-7 Segment Decoder

AN10

```

@DEVICE TYPE
82S153
@DRAWING
***** BINARY-TO-7 SEGMENT DECODER
@REVISION
***** REV. -
@DATE
***** OCT 1, 1984
@SYMBOL
***** FILE ID: 7deco
@COMPANY
***** SIGNETICS
@NAME
***** DAVID K. WONG
@DESCRIPTION
*****
* This circuit converts a 4-bit binary code ( HEX ) into
* a 7-segment display. The display is a common anode 7-segment LED.
* The output of the 82S153 goes LOW for each segment that is ON.
*****

          a
      f |-----| b
        | g |
      e |-----| c
          d

          TRUTH TABLE
          Z Y X W ----- a b c d e f g
          0 0 0 0      0 0 0 0 0 0 1
          0 0 0 1      1 0 0 1 1 1 1
          0 0 1 0      0 0 1 0 0 1 0
          0 0 1 1      0 0 0 0 1 1 0
          0 1 0 0      1 0 0 1 1 0 0
          0 1 0 1      0 1 0 0 1 0 0
          0 1 1 0      0 1 0 0 0 0 0
          0 1 1 1      0 0 0 1 1 1 1
          1 0 0 0      0 0 0 0 0 0 0
          1 0 0 1      0 0 0 0 1 0 0
          1 0 1 0      0 0 0 1 0 0 0
          1 0 1 1      1 1 0 0 0 0 0
          1 1 0 0      0 1 1 0 0 0 1
          1 1 0 1      1 0 0 0 1 0
          1 1 1 0      0 1 1 0 0 0 0
          1 1 1 1      0 1 1 1 0 0 0

@COMMON PRODUCT TERM
ZER = /z * /y * /x * /w ;
ONE = /z * /y * /x * w ;
TWO = /z * /y * x * /w ;
THR = /z * /y * x * w ;
FOU = /z * y * /x * /w ;
FIV = /z * y * /x * w ;
SIX = /z * y * x * /w ;
SEV = /z * y * x * w ;
EIG = z * /y * /x * /w ;
NIN = z * /y * /x * w ;
AAA = z * /y * x * /w ;
BBB = z * /y * x * w ;
CCC = z * y * /x * /w ;
DDD = z * y * /x * w ;
EEE = z * y * x * /w ;
FFF = z * y * x * w ;
    
```

T801020S

Figure 3. Boolean Equation for Binary-to-7 Segment Decoder

# 4-Bit Binary-To-7 Segment Decoder

AN10

```

@LOGIC EQUATION
/SEG_A = ONE + FOU + BBB + DDD ;
/SEG_B = FIV + SIX + BBB + CCC + EEE + FFF ;
/SEG_C = TWO + CCC + EEE + FFF ;
/SEG_D = ONE + FOU + SEV + AAA + FFF ;
/SEG_E = ONE + THR + FOU + FIV + SEV + NIN ;
/SEG_F = ONE + TWO + THR + SEV + DDD ;
/SEG_G = ZER + ONE + SEV + CCC ;

@I/O DIRECTION 150/1, 152/3, 154-9
"
*****
*          OUTPUTS ARE ENABLED WHEN /OE GOES LOW.          *
*          THEREFORE, D1...D7 = /( /OE) = OE                *
*****
D1 = OE ;
D2 = OE ;
D3 = OE ;
D4 = OE ;
D5 = OE ;
D6 = OE ;
D7 = OE ;

"
*****
*          END OF LOGIC EQUATIONS          *
*****
"

```

T801036S

Figure 3 (Continued)

In this example, an Active-Low output goes Low when a segment is to be turned on. The truth table shown in Table 1 is translated into an H/L programming table as shown in Table 2. Since the LED display used here is common anode, the output polarity of each output pin is programmed "L" so that an output goes low when a segment is to be turned on. The decoder takes 16 AND terms. Notice that all outputs go Low when input equals "8". If

the logic is written for "0s" instead of "1s", one AND term could be saved. Table 3 shows that the same circuit may be implemented by using only 15 AND terms. Notice that the polarity is the reverse of that of Table 2. The outputs may be turned on and off by I<sub>7</sub>, /OE, which controls the 7 tri-state output buffers through control terms (AND terms D<sub>1</sub> to D<sub>7</sub>). The same design is also implemented using Boolean equations with AMAZE. Figures 2

through 4 are, respectively, pin list, logic equations, and logic simulation results of the hex-to-7 segment decoder circuit generated by AMAZE. Table 4 is the H/L table generated by AMAZE. In the process of generating Boolean equations for the decoder, common product terms are used to economize the usage of AND terms.

# 4-Bit Binary-To-7 Segment Decoder

AN10

Table 4. H/L Table Generated by AMAZE for Binary-to-7 Segment Decoder

```

Cust/Project - ***** DAVID K. WONG
Date         - ***** OCT 1, 1984
Rev/I. D.    - ***** REV. -

82S153                ! POLARITY !
-----
T !                    !H:H:H:H:H:H:H:H:H!
E !                    -----
R !                    I      !      B(i)      !      B(o)      !
M !                    -----
  ! 7 6 5 4 3 2 1 0!9 8 7 6 5 4 3 2 1 0!9 8 7 6 5 4 3 2 1 0!
0! - - - ,L L L H! - - - - - - - - - - - - - - - !A A,A . . . A,A A A A!
1! - - - ,L H L L! - - - - - - - - - - - - - - - !A A,A . . . A,A . . . A!
2! - - - ,H L H H! - - - - - - - - - - - - - - - !A A,A . . . . . A!
3! - - - ,H H L H! - - - - - - - - - - - - - - - !A A,A . . . . . A!
4! - - - ,L H L H! - - - - - - - - - - - - - - - !A A, . A . . . A . . A!
5! - - - ,L H H L! - - - - - - - - - - - - - - - !A A, . A . . . . . A!
6! - - - ,H H L L! - - - - - - - - - - - - - - - !A A, . A . . . . . A!
7! - - - ,H H H L! - - - - - - - - - - - - - - - !A A, . A A . . . . A!
8! - - - ,H H H H! - - - - - - - - - - - - - - - !A A, . A A A . . . A!
9! - - - ,L L H L! - - - - - - - - - - - - - - - !A A, . . A . . . A . A!
10! - - - ,L H H H! - - - - - - - - - - - - - - - !A A, . . . A A A A A!
11! - - - ,H L L H! - - - - - - - - - - - - - - - !A A, . . . . A A A A!
12! - - - ,H L H L! - - - - - - - - - - - - - - - !A A, . . . . A, . . . A!
13! - - - ,L L H H! - - - - - - - - - - - - - - - !A A, . . . . . A A A!
14! - - - ,L L L L! - - - - - - - - - - - - - - - !A A, . . . . . A A!
15! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0! A A,A A A A A A A A!
16! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0! A A,A A A A A A A A!
17! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0! A A,A A A A A A A A!
18! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0! A A,A A A A A A A A!
19! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0! A A,A A A A A A A A!
20! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0! A A,A A A A A A A A!
21! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0! A A,A A A A A A A A!
22! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0! A A,A A A A A A A A!
23! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0! A A,A A A A A A A A!
24! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0! A A,A A A A A A A A!
25! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0! A A,A A A A A A A A!
26! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0! A A,A A A A A A A A!
27! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0! A A,A A A A A A A A!
28! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0! A A,A A A A A A A A!
29! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0! A A,A A A A A A A A!
30! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0! A A,A A A A A A A A!
31! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0! A A,A A A A A A A A!
D9! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0!
D8! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0!
D7! L - - - - - ! - - - - - ! - - - - - !
D6! L - - - - - ! - - - - - ! - - - - - !
D5! L - - - - - ! - - - - - ! - - - - - !
D4! L - - - - - ! - - - - - ! - - - - - !
D3! L - - - - - ! - - - - - ! - - - - - !
D2! L - - - - - ! - - - - - ! - - - - - !
D1! L - - - - - ! - - - - - ! - - - - - !
D0! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0 0 0! 0 0 0 0 0 0!

      / N N N Z Y X W N N / / / / / N N N / / / / / N
      O / / /      / / S S S S S S S / / / S S S S S S /
      E C C C      C C E E E E E E C C C E E E E E E C C
                    G G G G G G G      G G G G G G G
                    A B C D E F G      A B C D E F G
    
```

TB01040S

# 4-Bit Binary-To-7 Segment Decoder

AN10

```

825153 A:7deco153.STD
" 4-bit binary to 7-segment decoder simulation
"
" INPUTS <=B(I/O)=> TRACE TERMS
" 76543210 9876543210
"
00000000 ..LLLLLLH. ;
00000001 ..HLLHHH. ;
00000010 ..LHLLHL. ;
00000011 ..LLLLHHL. ;
00000100 ..HLLHLL. ;
00000101 ..LHLLHL. ;
00000110 ..LHLLLL. ;
00000111 ..LHLLHH. ;
00001000 ..LLLLLL. ;
00001001 ..LHLLHL. ;
00001010 ..LHLLLL. ;
00001011 ..LHLLHL. ;
00001100 ..LHLLHL. ;
00001101 ..LHLLHL. ;
00001110 ..LHLLHL. ;
00001111 ..LHLLHL. ;
10001111 ..LHLLHL. ;
10000000 ..... ;
"
" X----- I/O CONTROL LINES
" IIBBBBBBBI DESIGNATED I/O USAGE
" IIBBBBBBBI ACTUAL I/O USAGE
"
" PIN LIST...
" 08 07 06 05 04 03 02 01 19 18 17 16 15 14 13 12 11 09 ;
"
TB010505
    
```

Figure 4

## BCD-To-7 Segment Decoder

Using the same principle as in the previous example, a BCD input may be converted to a 7 segment display. If the input will always be within the range of 0000B to 1001B, the design will take ten AND terms to implement the decoding function. But if the display of an error message is desired when input exceeds 1001B, the design shown in Figures 5 and 6 and Table 5, may be used. Alternatively, the error message "E" will be displayed if the input does not equal anything between 0000B and 1001B inclusively. The circuit used to detect that none of the ten AND terms being true is shown in Figure 7 where if the input is outside of the range of 0000B to 1001B, none of the terms ZER to NIN will be active, which in turn causes the output of the NOR term to be High. The High output then causes the "E" term to be active and thus an output "E" is displayed. This scheme takes two propagation delays before "E" is displayed but only one delay for "0" to "9" to be displayed. The circuit shown in Figure 7 is implemented as shown in Table 4, column B(O)<sub>0</sub> of terms 0 to 9, and term #11. The output B(O)<sub>0</sub> becomes High when terms 0 to 9 are inactive, (T<sub>0</sub> + T<sub>1</sub> + T<sub>2</sub> + T<sub>3</sub> + T<sub>4</sub> + T<sub>5</sub> + T<sub>6</sub> + T<sub>7</sub> + T<sub>8</sub> + T<sub>9</sub>) = 0 where T<sub>0</sub> is term #0, T<sub>1</sub> is term #1, T<sub>2</sub> is . . . , etc. B(O)<sub>0</sub> is then fed back to the input of term #11 which, when active, causes outputs B and C to be High, segments B and C to turn off and the other segments to turn on. Term #10 is left blank to separate the main decoding function and the feedback function for the sake of clarity of this demonstration. A similar feature (the COMPLEMENT ARRAY) may be found in PLS155/157/159 where the feedback path is buried inside the chip without having to use up one I/O pin. Notice that this feature greatly reduces the number of AND terms needed. But for device architectures which do not allow common AND terms, this logic minimization will not be possible. Figure 8 and Table 6 implement design with AMAZE. The circuit is then simulated as shown in Figure 9.

```

***** P I N L I S T *****
L A B E L   ** F N C ** P I N   - - - - - P I N ** F N C **   L A B E L
W           ** I   ** 1-!       !-20 ** +SV **VCC
X           ** I   ** 2-!       !-19 ** B   **N/C
Y           ** I   ** 3-!       !-18 ** B   **N/C
Z           ** I   ** 4-!       8   !-17 ** B   **/SEG_A
N/C        ** I   ** 5-!       2   !-16 ** B   **/SEG_B
N/C        ** I   ** 6-!       9   !-15 ** B   **/SEG_C
N/C        ** I   ** 7-!       1   !-14 ** /B  **/SEG_D
/OE        ** I   ** 8-!       5   !-13 ** /B  **/SEG_E
N/C        ** B   ** 9-!       3   !-12 ** /B  **/SEG_F
GND        ** OV  ** 10-!      !-11 ** /B  **/SEG_G
    
```

Figure 5. Pin List of Binary-to-7 Segment Decoder Generated by AMAZE

# 4-Bit Binary-To-7 Segment Decoder

AN10

```

@DEVICE TYPE
82S153
@DRAWING
***** BCD-TO-7 SEGMENT DECODER
@REVISION
***** REV. -
@DATE
***** OCT 1, 1984
@SYMBOL
***** FILE ID: BCD-7153
@COMPANY
***** SIGNETICS
@NAME
***** DAVID K. WONG
@DESCRIPTION
*****
* This circuit converts a 4-bit BCD code into a 7-segment display. *
* The display is a common anode 7-segment LED. *
* The output of the 82S153 goes LOW for each segment that is ON. *
*****

```

TRUTH TABLE

		Z	Y	X	W	a	b	c	d	e	f	g
f	:	0	0	0	0	0	0	0	0	0	0	1
:	:	0	0	0	1	1	0	0	1	1	1	1
:	:	0	0	1	0	0	0	1	0	0	1	0
e	:	0	0	1	1	0	0	0	0	1	1	0
:	:	0	1	0	0	1	0	0	1	1	0	0
:	:	0	1	0	1	0	1	0	0	1	0	0
:	:	0	1	1	0	0	1	0	0	0	0	0
:	:	0	1	1	1	0	0	0	1	1	1	1
:	:	1	0	0	0	0	0	0	0	0	0	0
:	:	1	0	0	1	0	0	0	1	0	0	0
:	:	1	0	1	0	1	0	0	1	1	1	1
:	:	1	1	0	0	1	0	0	1	1	1	1
:	:	1	1	0	1	1	0	0	1	1	1	1
:	:	1	1	1	0	1	0	0	1	1	1	1
:	:	1	1	1	1	1	0	0	1	1	1	1

Display E (error)

```

@COMMON PRODUCT TERM
ZER = /z * /y * /x * /w ;
ONE = /z * /y * /x * w ;
TWO = /z * /y * x * /w ;
THR = /z * /y * x * w ;
FOU = /z * y * /x * /w ;
FIV = /z * y * /x * w ;
SIX = /z * y * x * /w ;
SEV = /z * y * x * w ;
EIG = z * /y * /x * /w ;
NIN = z * /y * /x * w ;
AAA = z * /y * x * /w ;
BBB = z * /y * x * w ;
CCC = z * y * /x * /w ;
DDD = z * y * /x * w ;
EEE = z * y * x * /w ;
FFF = z * y * x * w ;

```

Figure 6. Boolean Equations for BCD-to-7 Segment Decoder for AMAZ

TB010715





# 4-Bit Binary-To-7 Segment Decoder

AN10

```

@LOGIC EQUATION
/SEG_A = ONE + FOU + AAA
/SEG_B = FIV + SIX
/SEG_C = TWO
/SEG_D = /( ZER + TWO + THR
/SEG_E = /( ZER + TWO + SIX
/SEG_F = /( ZER + FOU + FIV
/SEG_G = /( TWO + THR + FOU

@I/O DIRECTION 150/1, 152/3,
"
*****
*          OUTPUTS ARE ENF
*          THEREFORE, D1..
*****
"
D1 = OE ;
D2 = OE ;
D3 = OE ;
D4 = OE ;
D5 = OE ;
D6 = OE ;
D7 = OE ;

"
*****
*          END
*****
"
    
```

Figure 6 (Continued)

TD010015

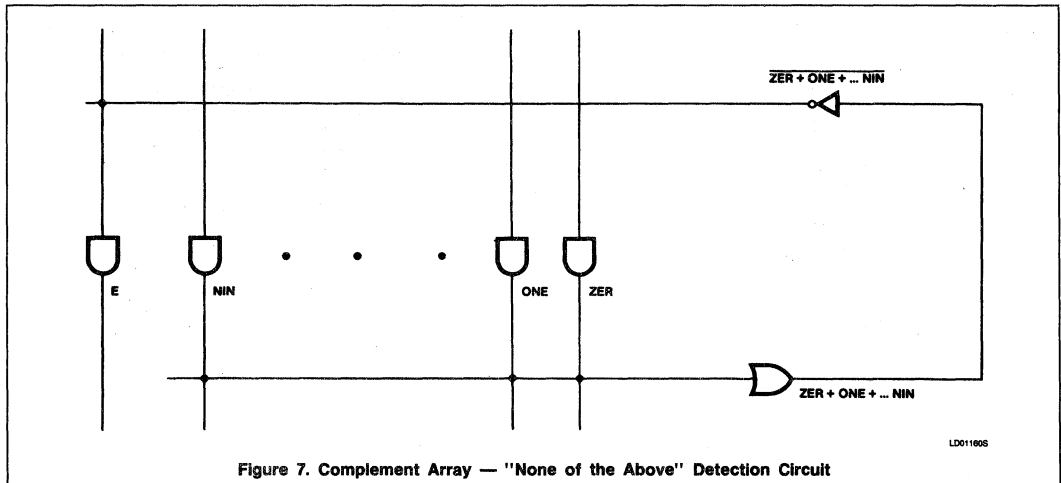


Figure 7. Complement Array — "None of the Above" Detection Circuit

LD010005

# 4-Bit Binary-To-7 Segment Decoder

AN10

Table 5. H/L Table for BCD-to-7 Segment Decoder Generated by AMAZE

```

Cust/Project - ***** DAVID K. WONG
Date         - ***** OCT 1, 1984
Rev/I. D.    - ***** REV. -

B2S153                                !    POLARITY    !
-----
T !                                     !H:L:H:L:H:L:L:L:H!
E !
R !          I          !          B(i)          !          B(o)          !
M !
-----
! 7 6 5 4 3 2 1 0!9 8 7 6 5 4 3 2 1 0!9 8 7 6 5 4 3 2 1 0!
0!  - - - ,L L L H! - - - - - - - - - - - - - - - !A A,A . . . . A!
1!  - - - ,L H L L! - - - - - - - - - - - - - - - !A A,A . . . . A A A!
2!  - - - ,H L H L! - - - - - - - - - - - - - - - !A A,A . . . . A A!
3!  - - - ,H L H H! - - - - - - - - - - - - - - - !A A,A . . . . A!
4!  - - - ,H H L L! - - - - - - - - - - - - - - - !A A,A . . . . A!
5!  - - - ,H H L H! - - - - - - - - - - - - - - - !A A,A . . . . A!
6!  - - - ,H H H L! - - - - - - - - - - - - - - - !A A,A . . . . A!
7!  - - - ,H H H H! - - - - - - - - - - - - - - - !A A,A . . . . A!
8!  - - - ,L H L H! - - - - - - - - - - - - - - - !A A,. A . A. A A A!
9!  - - - ,L H H L! - - - - - - - - - - - - - - - !A A,. A . A. A A A!
10! - - - ,L L H L! - - - - - - - - - - - - - - - !A A,. A . A. A A A!
11! - - - ,L L L L! - - - - - - - - - - - - - - - !A A,. . . A. A A A!
12! - - - ,L L H H! - - - - - - - - - - - - - - - !A A,. . . A. A A A!
13! - - - ,H L L L! - - - - - - - - - - - - - - - !A A,. . . A. A A A!
14! - - - ,H L L H! - - - - - - - - - - - - - - - !A A,. . . A. A A A!
15!0 0 0 0 0 0 0 0!0 0 0 0 0 0 0 0 0 0 0 0!A A,A A A A,A A A A!
16!0 0 0 0 0 0 0 0!0 0 0 0 0 0 0 0 0 0 0 0!A A,A A A A,A A A A!
17!0 0 0 0 0 0 0 0!0 0 0 0 0 0 0 0 0 0 0 0!A A,A A A A,A A A A!
18!0 0 0 0 0 0 0 0!0 0 0 0 0 0 0 0 0 0 0 0!A A,A A A A,A A A A!
19!0 0 0 0 0 0 0 0!0 0 0 0 0 0 0 0 0 0 0 0!A A,A A A A,A A A A!
20!0 0 0 0 0 0 0 0!0 0 0 0 0 0 0 0 0 0 0 0!A A,A A A A,A A A A!
21!0 0 0 0 0 0 0 0!0 0 0 0 0 0 0 0 0 0 0 0!A A,A A A A,A A A A!
22!0 0 0 0 0 0 0 0!0 0 0 0 0 0 0 0 0 0 0 0!A A,A A A A,A A A A!
23!0 0 0 0 0 0 0 0!0 0 0 0 0 0 0 0 0 0 0 0!A A,A A A A,A A A A!
24!0 0 0 0 0 0 0 0!0 0 0 0 0 0 0 0 0 0 0 0!A A,A A A A,A A A A!
25!0 0 0 0 0 0 0 0!0 0 0 0 0 0 0 0 0 0 0 0!A A,A A A A,A A A A!
26!0 0 0 0 0 0 0 0!0 0 0 0 0 0 0 0 0 0 0 0!A A,A A A A,A A A A!
27!0 0 0 0 0 0 0 0!0 0 0 0 0 0 0 0 0 0 0 0!A A,A A A A,A A A A!
28!0 0 0 0 0 0 0 0!0 0 0 0 0 0 0 0 0 0 0 0!A A,A A A A,A A A A!
29!0 0 0 0 0 0 0 0!0 0 0 0 0 0 0 0 0 0 0 0!A A,A A A A,A A A A!
30!0 0 0 0 0 0 0 0!0 0 0 0 0 0 0 0 0 0 0 0!A A,A A A A,A A A A!
31!0 0 0 0 0 0 0 0!0 0 0 0 0 0 0 0 0 0 0 0!A A,A A A A,A A A A!
D9!0 0 0 0 0 0 0 0!0 0 0 0 0 0 0 0 0 0 0 0!
D8!0 0 0 0 0 0 0 0!0 0 0 0 0 0 0 0 0 0 0 0!
D7!L - - - - - - - - - - - - - - - - - - - - - !
D6!L - - - - - - - - - - - - - - - - - - - - - !
D5!L - - - - - - - - - - - - - - - - - - - - - !
D4!L - - - - - - - - - - - - - - - - - - - - - !
D3!L - - - - - - - - - - - - - - - - - - - - - !
D2!L - - - - - - - - - - - - - - - - - - - - - !
D1!L - - - - - - - - - - - - - - - - - - - - - !
D0!0 0 0 0 0 0 0 0!0 0 0 0 0 0 0 0 0 0 0 0!

/ N N N Z Y X W N N / / / / / / N N N / / / / / / N
0 / / / / / S S S S S S S / / / S S S S S S S /
E C C C C C E E E E E E E C C C C E E E E E E E C
          G G G G G G G G          G G G G G G G G
          A B C D E F G          A B C D E F G
    
```

TB010905

# 4-Bit Binary-To-7 Segment Decoder

AN10

```

@DEVICE TYPE
***** B2S153
@DRAWING
***** BINARY-TO-7 SEGMENT DECODER
@REVISION
***** REV. -
@DATE
***** OCT 1, 1984
@SYMBOL
***** FILE ID: BCD77153
@COMPANY
***** SIGNETICS
@NAME
***** DAVID K. WONG
@DESCRIPTION
*****
* This circuit converts a 4-bit BCD code into a 7-segment display. *
* The display is a common anode 7-segment LED. *
* The output of the B2S153 goes LOW for each segment that is ON. *
*****

                                TRUTH TABLE
                                a b c d e f g
-----
a      Z Y X W
-----
f |      | b      0 0 0 0      0 0 0 0 0 0 1
 | g |      0 0 0 1      1 0 0 1 1 1 1
-----
e |      | c      0 0 1 0      0 0 1 0 0 1 0
 |      |      0 0 1 1      0 0 0 0 1 1 0
 |      |      0 1 0 0      1 0 0 1 1 0 0
-----
d      0 1 0 1      0 1 0 0 1 0 0
      0 1 1 0      0 1 0 0 0 0 0
      0 1 1 1      0 0 0 1 1 1 1
      1 0 0 0      0 0 0 0 0 0 0
      1 0 0 1      0 0 0 0 1 0 0

Over Range      1 0 1 0      0 1 1 0 0 0 0      Display E (error)
"                1 0 1 1      0 1 1 0 0 0 0      "
"                1 1 0 0      0 1 1 0 0 0 0      "
"                1 1 0 1      0 1 1 0 0 0 0      "
"                1 1 1 0      0 1 1 0 0 0 0      "
"                1 1 1 1      0 1 1 0 0 0 0      "

@COMMON PRODUCT TERM
ZER = /z * /y * /x * /w ;
ONE = /z * /y * /x * w ;
TWO = /z * /y * x * /w ;
THR = /z * /y * x * w ;
FOU = /z * y * /x * /w ;
FIV = /z * y * /x * w ;
SIX = /z * y * x * /w ;
SEV = /z * y * x * w ;
EIG = z * /y * /x * /w ;
NIN = z * /y * /x * w ;
    
```

T801101S

Figure 8. Boolean Equation for BCD-to-7 Segment Decoder

## 4-Bit Binary-To-7 Segment Decoder

AN10

## @LOGIC EQUATION

```

"
*****
* When input is greater than 9, OVR ( Over Range ) will go HIGH, WHICH *
* will be fed-back to the AND array to control the segments.          *
* Here the OVR pin and the feedback loop is used as COMPLEMENT ARRAY. *
* The COMPLEMENT ARRAY saves 5 P-terms but loses one I/O pin.       *
*****
"

OVR = /( ZER + ONE + TWO + THR + FOU + FIV + SIX + SEV + EIG + NIN ) ;
/SEG_A = ONE + FOU ;
/SEG_B = FIV + SIX + OVR ;
/SEG_C = TWO + OVR ;
/SEG_D = ONE + FOU + SEV ;
/SEG_E = /( ZER + TWO + SIX + EIG + OVR ) ;
/SEG_F = ONE + TWO + THR + SEV ;
/SEG_G = ZER + ONE + SEV ;

@I/O DIRECTION 150/1, 152/3, 154-9
"
*****
* OUTPUTS ARE ENABLED WHEN /OE GOES LOW.                               *
* THEREFORE, D1..D7 = /( /OE ) = OE ;                                  *
*****
"
D1 = OE ;
D2 = OE ;
D3 = OE ;
D4 = OE ;
D5 = OE ;
D6 = OE ;
D7 = OE ;

"
*****
* END OF LOGIC EQUATIONS                                               *
*****
"

```

TB01111S

Figure 8 (Continued)

# 4-Bit Binary-To-7 Segment Decoder

AN10

Table 6. H/L Table of BCD-to-7 Segment Decoder Generated by AMAZE

```

Cust/Project - ***** DAVID K. WONG
Date         - ***** OCT 1, 1984
Rev/I. D.    - ***** REV. -

82S153                                     ! POLARITY !
-----
T !                                         !H:LH:LH:LH:LH:LH:L!
E !
R !         I         !         B(i)         !         B(o)         !
M !
-----
!7_6_5_4_3_2_1_0!9_8_7_6_5_4_3_2_1_0!9_8_7_6_5_4_3_2_1_0!
0!- - - ,L L L L!- - - , - - - , - - - !A A, . . . ,A . A A!
1!- - - ,L L L H!- - - , - - - , - - - !A A,A . . A, A A A!
2!- - - ,L L H L!- - - , - - - , - - - !A A, . . . ,A A A!
3!- - - ,L L H H!- - - , - - - , - - - !A A, . . . ,A . A!
4!- - - ,L H L L!- - - , - - - , - - - !A A,A . . A, . . A!
5!- - - ,L H L H!- - - , - - - , - - - !A A, A . . . . A!
6!- - - ,L H H L!- - - , - - - , - - - !A A, . A . . . ,A . A!
7!- - - ,L H H H!- - - , - - - , - - - !A A, . . . A, A A A!
8!- - - ,H L L L!- - - , - - - , - - - !A A, . . . ,A . . A!
9!- - - ,H L L H!- - - , - - - , - - - !A A, . . . . A!
10!- - - , - - - !- - - , - - - , - - - H!A A, . A . A . . !
11!0 0 0 0,0 0 0!0 0 0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
12!0 0 0 0,0 0 0!0 0 0 0,0 0 0 0,0 0 0 0!A A, A A A A,A A A A!
13!0 0 0 0,0 0 0!0 0 0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
14!0 0 0 0,0 0 0!0 0 0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
15!0 0 0 0,0 0 0!0 0 0 0,0 0 0 0,0 0 0 0!A A, A A A A,A A A A!
16!0 0 0 0,0 0 0!0 0 0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
17!0 0 0 0,0 0 0!0 0 0 0,0 0 0 0,0 0 0 0!A A, A A A A,A A A A!
18!0 0 0 0,0 0 0!0 0 0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
19!0 0 0 0,0 0 0!0 0 0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
20!0 0 0 0,0 0 0!0 0 0 0,0 0 0 0,0 0 0 0!A A, A A A A,A A A A!
21!0 0 0 0,0 0 0!0 0 0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
22!0 0 0 0,0 0 0!0 0 0 0,0 0 0 0,0 0 0 0!A A, A A A A,A A A A!
23!0 0 0 0,0 0 0!0 0 0 0,0 0 0 0,0 0 0 0!A A,A A A A,A A A A!
24!0 0 0 0,0 0 0!0 0 0 0,0 0 0 0,0 0 0 0!A A, A A A A,A A A A!
25!0 0 0 0,0 0 0!0 0 0 0,0 0 0 0,0 0 0 0!A A, A A A A,A A A A!
26!0 0 0 0,0 0 0!0 0 0 0,0 0 0 0,0 0 0 0!A A, A A A A,A A A A!
27!0 0 0 0,0 0 0!0 0 0 0,0 0 0 0,0 0 0 0!A A, A A A A,A A A A!
28!0 0 0 0,0 0 0!0 0 0 0,0 0 0 0,0 0 0 0!A A, A A A A,A A A A!
29!0 0 0 0,0 0 0!0 0 0 0,0 0 0 0,0 0 0 0!A A, A A A A,A A A A!
30!0 0 0 0,0 0 0!0 0 0 0,0 0 0 0,0 0 0 0!A A, A A A A,A A A A!
31!0 0 0 0,0 0 0!0 0 0 0,0 0 0 0,0 0 0 0!A A, A A A A,A A A A!
D9!0 0 0 0,0 0 0!0 0 0 0,0 0 0 0,0 0 0 0!
D8!0 0 0 0,0 0 0!0 0 0 0,0 0 0 0,0 0 0 0!
D7!L - - - , - - - !- - - , - - - , - - - !
D6!L - - - , - - - !- - - , - - - , - - - !
D5!L - - - , - - - !- - - , - - - , - - - !
D4!L - - - , - - - !- - - , - - - , - - - !
D3!L - - - , - - - !- - - , - - - , - - - !
D2!L - - - , - - - !- - - , - - - , - - - !
D1!L - - - , - - - !- - - , - - - , - - - !
D0!- - - , - - - !- - - , - - - , - - - !

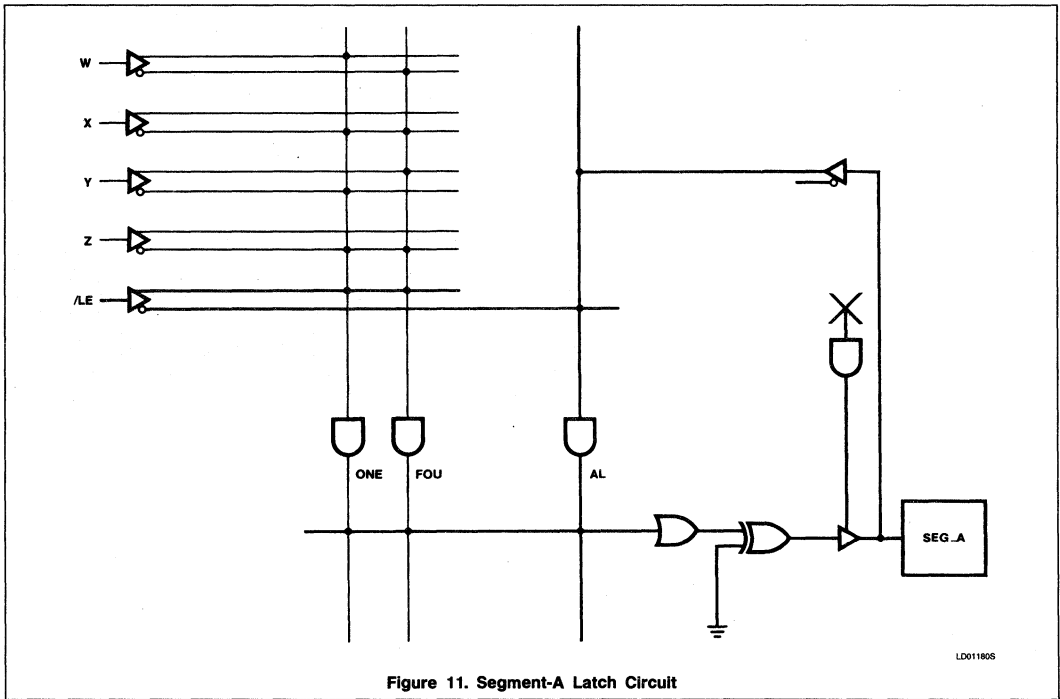
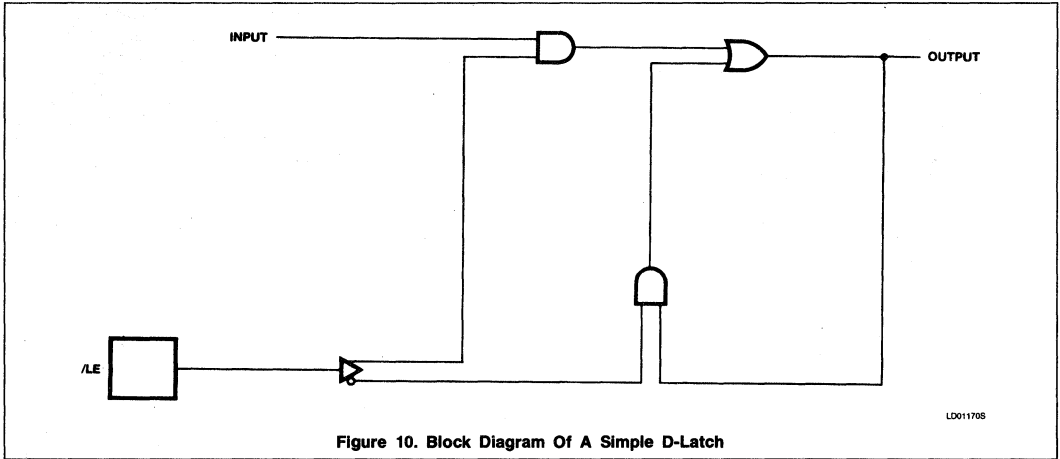
/ N N N Z Y X W N N / / / / / / D N N / / / / / / O
0 / / / / / S S S S S S S V / / S S S S S S S S V
E C C C C C C E E E E E E E R C C E E E E E E E R
G G G G G G G G G G G G G G G G G G G G G G
A B C D E F G A B C D E F G
    
```

TB011205



# 4-Bit Binary-To-7 Segment Decoder

AN10



# 4-Bit Binary-To-7 Segment Decoder

AN10

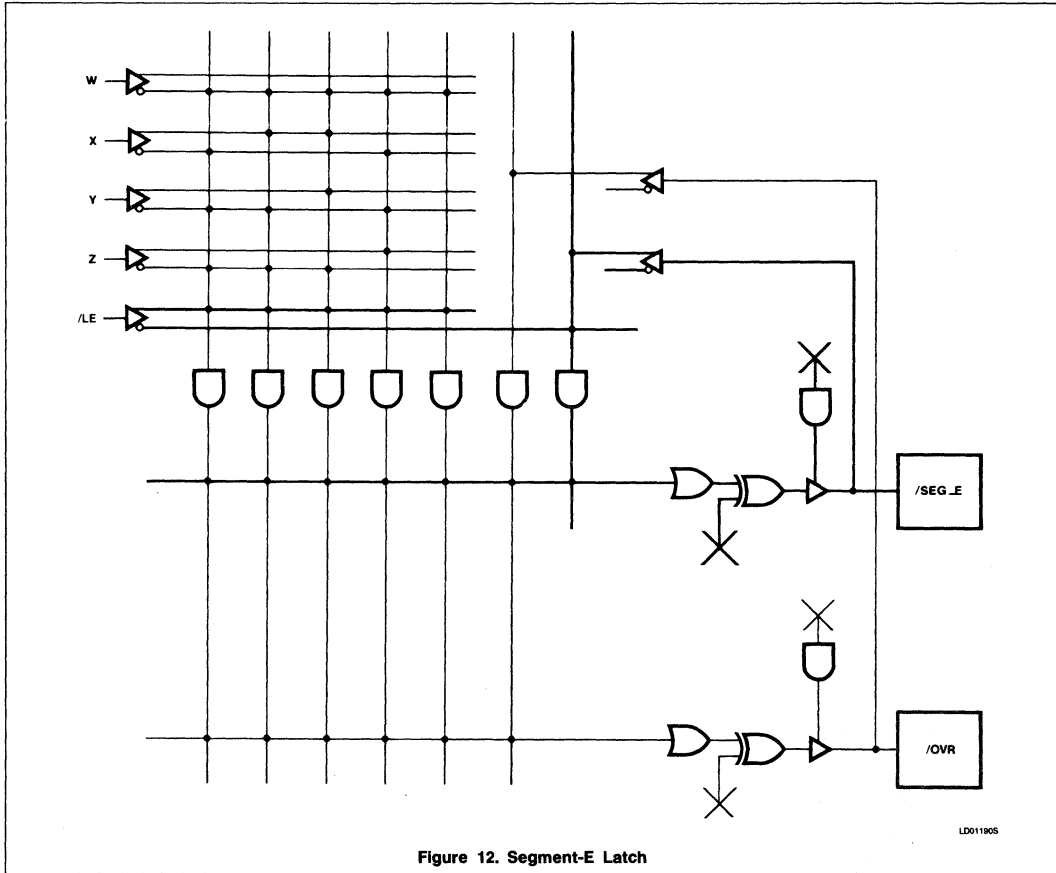


Figure 12. Segment-E Latch

LD011905

\*\*\*\*\* PIN LIST \*\*\*\*\*

LABEL	** FNC	**PIN	PIN**	FNC **	LABEL
W	** I	** 1-1	1-20	** +5V	**VCC
X	** I	** 2-1	1-19	** B	**N/C
Y	** I	** 3-1	1-18	** B	**N/C
Z	** I	** 4-1	8	1-17	** 0 **/SEG_A
N/C	** I	** 5-1	2	1-16	** 0 **/SEG_B
N/C	** I	** 6-1	5	1-15	** 0 **/SEG_C
/LE	** I	** 7-1	1	1-14	** 0 **/SEG_D
N/C	** I	** 8-1	5	1-13	** /0 **/SEG_E
OVR	** /0	** 9-1	3	1-12	** 0 **/SEG_F
GND	** OV	** 10-1		1-11	** 0 **/SEG_G

TD011405

Figure 13. Pin List of BCD-to-7 Segment Decoder with Latches



# 4-Bit Binary-To-7 Segment Decoder

AN10

```

@DEVICE TYPE
82S153
@DRAWING
***** BCD-TO-7 SEGMENT DECODER
@REVISION
***** REV. -
@DATE
***** OCT 1, 1984
@SYMBOL
***** FILE ID: BCD7L153
@COMPANY
***** SIGNETICS
@NAME
***** DAVID K. WONG
@DESCRIPTION
*****
* This circuit converts a 4-bit BCD code into a 7-segment display. *
* The display is a common anode 7-segment LED. *
* The output of the 82S153 goes LOW for each segment that is ON. *
* The outputs are latched when /LE goes LOW. *
*****

          a          TRUTH TABLE
          |          |
          | Z Y X W |-----| a b c d e f g
          |-----|-----|-----|-----|-----|
f |   |   | b |
 |---|   |   |
 | g |   |   |
e |   |   | c |
 |---|   |   |
 |   |   |   |
          | d |
          |---|
          | 1 0 1 0 | 0 0 0 0 | 0 0 0 0 0 0 1
          | 1 0 0 1 | 0 0 0 1 | 1 0 0 1 1 1 1
          | 0 0 1 0 | 0 0 1 0 | 0 0 1 0 0 1 0
          | 0 0 1 1 | 0 0 1 1 | 0 0 0 0 1 1 0
          | 0 1 0 0 | 0 1 0 0 | 1 0 0 1 1 0 0
          | 0 1 0 1 | 0 1 0 1 | 0 1 0 0 1 0 0
          | 0 1 1 0 | 0 1 1 0 | 0 1 0 0 0 0 0
          | 0 1 1 1 | 0 0 0 1 | 0 0 0 1 1 1 1
          | 1 0 0 0 | 0 0 0 0 | 0 0 0 0 0 0 0
          | 1 0 0 1 | 0 0 0 0 | 0 0 0 0 1 0 0

          Over Range 1 0 1 0 | 0 1 1 0 0 0 0 Display E (error)
          "           1 0 1 1 | 0 1 1 0 0 0 0 "
          "           1 1 0 0 | 0 1 1 0 0 0 0 "
          "           1 1 0 1 | 0 1 1 0 0 0 0 "
          "           1 1 1 0 | 0 1 1 0 0 0 0 "
          "           1 1 1 1 | 0 1 1 0 0 0 0 "

@COMMON PRODUCT TERM
ZER = /z * /y * /x * /w * /LE ;
ONE  = /z * /y * /x * w * /LE ;
TWO  = /z * /y * x * /w * /LE ;
THR  = /z * /y * x * w * /LE ;
FOU  = /z * y * /x * /w * /LE ;
FIV  = /z * y * /x * w * /LE ;
SIX  = /z * y * x * /w * /LE ;
SEV  = /z * y * x * w * /LE ;
EIG  = z * /y * /x * /w * /LE ;
NIN  = z * /y * /x * w * /LE ;

"
*****
* Feedback circuits to form D-latches. *
*****

```

TBD11505

Figure 14. Boolean Equation of BCD-to-7 Segment Decoder with Latches

## 4-Bit Binary-To-7 Segment Decoder

AN10

```

"
AL = LE * /SEG_A ;
BL = LE * /SEG_B ;
CL = LE * /SEG_C ;
DL = LE * /SEG_D ;
EL = LE * SEG_E ; "Note that in LOGIC EQUATION definition, SEG_E is
                    inverted. Therefore, the feedback for E has to
                    be inverted too."
FL = LE * /SEG_F ;
GL = LE * /SEG_G ;

@LOGIC EQUATION

"
*****
* When input is greater than 9, OVR ( Over Range ) will go HIGH, WHICH *
* will be fed-back to the AND array to control the segments. *
* Here the OVR pin and the feedback loop is used as COMPLEMENT ARRAY. *
* The COMPLEMENT ARRAY saves 5 P-terms but uses up one I/O pin. *
*****

OVR = / ( ZER + ONE + TWO + THR + FOU + FIV + SIX + SEV + EIG + NIN + LE ) ;
/SEG_A = ONE + FOU + AL ;
/SEG_B = FIV + SIX + OVR + BL ;
/SEG_C = TWO + OVR + CL ;
/SEG_D = ONE + FOU + SEV + DL ;
/SEG_E = / ( ZER + TWO + SIX + EIG + OVR + EL ) ;
/SEG_F = ONE + TWO + THR + SEV + FL ;
/SEG_G = ZER + ONE + SEV + GL ;

@I/O DIRECTION
"
*****
* END OF LOGIC EQUATIONS
*****
"

```

TB01160S

Figure 14 (Continued)

# 4-Bit Binary-To-7 Segment Decoder

AN10

Table 7. H/L Table of BCD-to-7 Segment Decoder with Latches

Cust/Project - \*\*\*\*\* DAVID K. WONG  
 Date - \*\*\*\*\* OCT 1, 1984  
 Rev/I. D. - \*\*\*\*\* REV. -

82S153		POLARITY																			
T		!H:H:H:H:H:H:L:H:H:L!																			
E		-----																			
R		I	B(i)			B(o)															
M		7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0		
0!	- H -	-,L	L	L	L	!	-	-	-	-	!	A	A,	,	,,	A	,	A	!		
1!	- H -	-,L	L	L	H	!	-	-	-	-	!	A	A,	A	,	A,	A	A	!		
2!	- H -	-,L	L	H	L	!	-	-	-	-	!	A	A,	,	,	A	,	A	!		
3!	- H -	-,L	L	H	H	!	-	-	-	-	!	A	A,	,	,	,	A	,	A	!	
4!	- H -	-,L	H	L	L	!	-	-	-	-	!	A	A,	A	,	,	,	,	A	!	
5!	- H -	-,L	H	L	H	!	-	-	-	-	!	A	A,	A	,	,	,	,	A	!	
6!	- H -	-,L	H	H	L	!	-	-	-	-	!	A	A,	,	,	,	,	,	A	!	
7!	- H -	-,L	H	H	H	!	-	-	-	-	!	A	A,	,	,	A	,	,	A	!	
8!	- H -	-,H	L	L	L	!	-	-	-	-	!	A	A,	,	,	,	,	,	A	!	
9!	- H -	-,H	L	L	H	!	-	-	-	-	!	A	A,	,	,	,	,	,	A	!	
10!	- L -	-	-	-	-	!	-	-	-	-	!	A	A,	,	,	,	,	,	A	!	
11!	- L -	-	-	-	-	!	H	-	-	-	!	A	A,	A	,	,	,	,	A	!	
12!	-	-	-	-	-	!	-	-	-	-	!	H	A	,	A	,	A	,	A	!	
13!	- L -	-	-	-	-	!	-	H	-	-	!	A	A,	,	,	,	,	,	A	!	
14!	- L -	-	-	-	-	!	-	-	H	-	!	A	A,	,	,	A	,	,	,	A	!
15!	- L -	-	-	-	-	!	-	-	-	H	!	A	A,	,	,	A	,	,	,	A	!
16!	- L -	-	-	-	-	!	-	-	-	L	!	A	A,	,	,	,	,	,	,	A	!
17!	- L -	-	-	-	-	!	-	-	-	-	H	!	A	A,	,	,	,	,	,	A	!
18!	- L -	-	-	-	-	!	-	-	-	-	H	!	A	A,	,	,	,	,	,	A	!
19!	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	A	A	A	A	A
20!	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	A	A	A	A	A
21!	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	A	A	A	A	A
22!	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	A	A	A	A	A
23!	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	A	A	A	A	A
24!	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	A	A	A	A	A
25!	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	A	A	A	A	A
26!	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	A	A	A	A	A
27!	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	A	A	A	A	A
28!	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	A	A	A	A	A
29!	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	A	A	A	A	A
30!	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	A	A	A	A	A
31!	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	A	A	A	A	A
D9!	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D8!	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D7!	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
D6!	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
D5!	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
D4!	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
D3!	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
D2!	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
D1!	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
DO!	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

N	/	N	N	Z	Y	X	W	N	N	/	/	/	/	/	0	N	/	/	/	/	/	0				
/	L	/	/	/	/	/	/	/	S	S	S	S	S	S	S	V	/	/	S	S	S	S	S	S	S	V
C	E	C	C	C	C	C	C	E	E	E	E	E	E	E	E	R	C	C	E	E	E	E	E	E	E	R
								G	G	G	G	G	G	G	G											
								A	B	C	D	E	F	G												

TB011705

# 4-Bit Binary-To-7 Segment Decoder

AN10

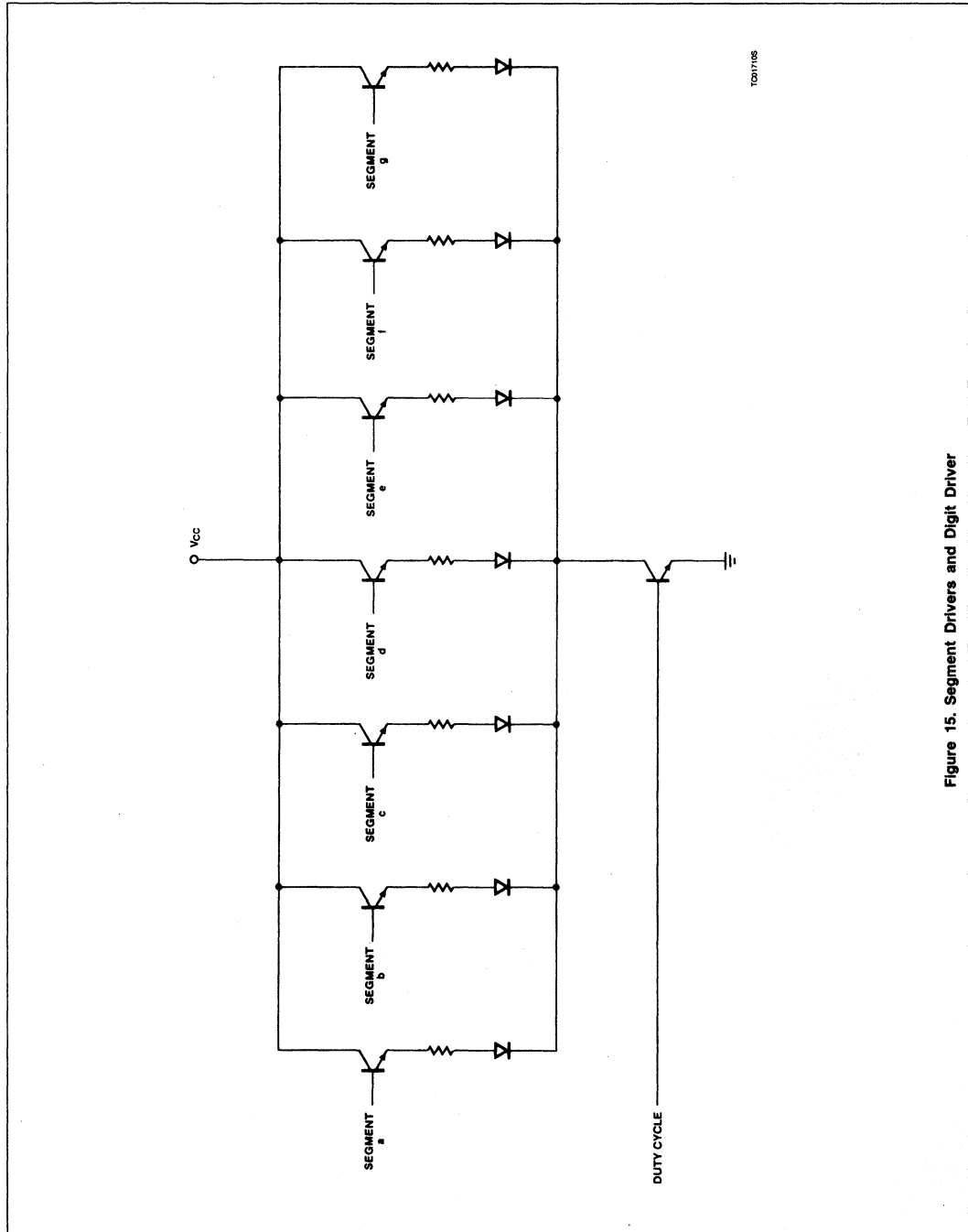


Figure 15. Segment Drivers and Digit Driver

# 4-Bit Binary-To-7 Segment Decoder

AN10

**Table 8. BCD-to-7 Segment Decoder Using Complement Array**

82S153/153A PROGRAMMING TABLE      DRAWING ID: BCD TO 7SEG DECODE REV. —  
 DESIGNER: D.K. Wang      DATE: 12/15/84      POLARITY

	AND							OR						
	BCD			BCD				BCD			BCD			
	7:6:5:4	3:2:1:0	9:8	7:6:5:4	3:2:1:0	9:8	7:6:5:4	3:2:1:0	9:8	7:6:5:4	3:2:1:0	9:8	7:6:5:4	3:2:1:0
0	---	---	---	---	---	---	---	---	---	---	---	---	---	---
1	---	---	---	---	---	---	---	---	---	---	---	---	---	---
2	---	---	---	---	---	---	---	---	---	---	---	---	---	---
3	---	---	---	---	---	---	---	---	---	---	---	---	---	---
4	---	---	---	---	---	---	---	---	---	---	---	---	---	---
5	---	---	---	---	---	---	---	---	---	---	---	---	---	---
6	---	---	---	---	---	---	---	---	---	---	---	---	---	---
7	---	---	---	---	---	---	---	---	---	---	---	---	---	---
8	---	---	---	---	---	---	---	---	---	---	---	---	---	---
9	---	---	---	---	---	---	---	---	---	---	---	---	---	---
10	---	---	---	---	---	---	---	---	---	---	---	---	---	---
11	---	---	---	---	---	---	---	---	---	---	---	---	---	---
12	---	---	---	---	---	---	---	---	---	---	---	---	---	---
13	---	---	---	---	---	---	---	---	---	---	---	---	---	---
14	---	---	---	---	---	---	---	---	---	---	---	---	---	---
15	---	---	---	---	---	---	---	---	---	---	---	---	---	---
16	---	---	---	---	---	---	---	---	---	---	---	---	---	---
17	---	---	---	---	---	---	---	---	---	---	---	---	---	---
18	---	---	---	---	---	---	---	---	---	---	---	---	---	---
19	---	---	---	---	---	---	---	---	---	---	---	---	---	---
20	---	---	---	---	---	---	---	---	---	---	---	---	---	---
21	---	---	---	---	---	---	---	---	---	---	---	---	---	---
22	---	---	---	---	---	---	---	---	---	---	---	---	---	---
23	---	---	---	---	---	---	---	---	---	---	---	---	---	---
24	---	---	---	---	---	---	---	---	---	---	---	---	---	---
25	---	---	---	---	---	---	---	---	---	---	---	---	---	---
26	---	---	---	---	---	---	---	---	---	---	---	---	---	---
27	---	---	---	---	---	---	---	---	---	---	---	---	---	---
28	---	---	---	---	---	---	---	---	---	---	---	---	---	---
29	---	---	---	---	---	---	---	---	---	---	---	---	---	---
30	---	---	---	---	---	---	---	---	---	---	---	---	---	---
31	---	---	---	---	---	---	---	---	---	---	---	---	---	---
D9	---	---	---	---	---	---	---	---	---	---	---	---	---	---
D8	---	---	---	---	---	---	---	---	---	---	---	---	---	---
D7	L	---	---	---	---	---	---	---	---	---	---	---	---	---
D6	L	---	---	---	---	---	---	---	---	---	---	---	---	---
D5	L	---	---	---	---	---	---	---	---	---	---	---	---	---
D4	L	---	---	---	---	---	---	---	---	---	---	---	---	---
D3	L	---	---	---	---	---	---	---	---	---	---	---	---	---
D2	L	---	---	---	---	---	---	---	---	---	---	---	---	---
D1	L	---	---	---	---	---	---	---	---	---	---	---	---	---
D0	L	---	---	---	---	---	---	---	---	---	---	---	---	---
N	---	---	---	---	---	---	---	---	---	---	---	---	---	---
A	---	---	---	---	---	---	---	---	---	---	---	---	---	---
H	---	---	---	---	---	---	---	---	---	---	---	---	---	---
E	---	---	---	---	---	---	---	---	---	---	---	---	---	---

*(\*) NOTICE THAT IN THIS PARTICULAR APPLICATION, THIS TERM MUST BE PRESENT FOR THE COMPLEMENT ARRAY TO FUNCTION PROPERLY.*

TB011805

# 4-Bit Binary-To-7 Segment Decoder

AN10

Table 9. Segments A and E with Latches

82S153/153A PROGRAMMING TABLE DRAWING ID: \_\_\_\_\_ REV. \_\_\_\_\_  
 DESIGNER: \_\_\_\_\_ DATE: \_\_\_\_\_

DEC	A(0)							B(1)							POLARITY							OR							
	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	1	2	3		4	5	6	7	8	9	0
0	H	-	-	-	L	L	L	H	-	-	-	-	-	-	-	-	-	-	A										$\overline{A} = \overline{LE} * \overline{WR} * \overline{Y} * \overline{E}$
1	H	-	-	-	L	H	L	L	-	-	-	-	-	-	-	-	-	-	A										$+ \overline{LE} * \overline{WR} * \overline{Y} * \overline{E}$
2	L	-	-	-	-	-	-	-	H	-	-	-	-	-	-	-	-	-	A										$+ \overline{LE} * \overline{E}$
3																													
4																													
5	H	-	-	-	L	L	L	L	-	-	-	-	-	-	-	-	-	-	A										$E = \overline{LE} * \overline{WR} * \overline{Y} * \overline{E}$
6	H	-	-	-	L	L	H	L	-	-	-	-	-	-	-	-	-	-	A										$+ \overline{LE} * \overline{WR} * \overline{Y} * \overline{E}$
7	H	-	-	-	L	H	H	L	-	-	-	-	-	-	-	-	-	-	A										$+ \overline{LE} * \overline{WR} * \overline{Y} * \overline{E}$
8	H	-	-	-	H	L	L	L	-	-	-	-	-	-	-	-	-	-	A										$+ \overline{LE} * \overline{WR} * \overline{Y} * \overline{E}$
9	H	-	-	-	-	-	-	-	H	-	-	-	-	-	-	-	-	-	A										$+ \overline{LE} * \overline{E}$
10	L	-	-	-	-	-	-	-	L	-	-	-	-	-	-	-	-	-	A										$+ \overline{LE} * \overline{E}$
11																													
12																													
13																													
14																													
15																													
16																													
17																													
18																													
19																													
20																													
21																													
22																													
23																													
24																													
25																													
26																													
27																													
28																													
29																													
30																													
31																													
D9																													
D8																													
D7																													
D6																													
D5																													
D4																													
D3																													
D2																													
D1																													
D0																													
N																													
A																													
E																													
H																													
E																													
OVER																													

T801191S

# 4-Bit Binary-To-7 Segment Decoder

AN10

Table 10. H/L Table of BCD-to-7 Segment Decoder with Latches with Outputs Active-High

82S153/153A PROGRAMMING TABLE     DRAWING ID: BCD7LM     REV. --  
DESIGNER: D. C. WONG     DATE: 12/20/84

	AND							OR										
	A(0)				B(1)			C(0)										
	7:6:5:4	3:2:1:0	9:8	7:6:5:4	3:2:1:0	9:8	7:6:5:4	3:2:1:0										
0	-H:-	L:L:L:L					A:A:A:A	A										0
1	-H:-	L:L:L:H					A:A:A:A	A										1
2	-H:-	L:L:H:L					A:A:A:A	A										2
3	-H:-	L:L:H:H					A:A:A:A	A										3
4	-H:-	L:H:L:L					A:A:A:A	A										4
5	-H:-	L:H:L:H					A:A:A:A	A										5
6	-H:-	L:H:H:L					A:A:A:A	A										6
7	-H:-	L:H:H:H					A:A:A:A	A										7
8	-H:-	H:L:L:L					A:A:A:A	A										8
9	-H:-	H:L:L:H					A:A:A:A	A										9
10																		
11																		
12									A:A:									E (ERROR)
13	-L-								A:?									SEG A LATCH
14	-L-								A:?									" B "
15	-L-								A:?									" C "
16	-L-								A:?									" D "
17	-L-								A:?									" E "
18	-L-								A:?									" F "
19	-L-								A:?									" G "
20																		
21																		
22																		
23																		
24																		
25																		
26																		
27																		
28																		
29																		
30																		
31																		
D9																		
D8																		
D7																		
D6																		
D5																		
D4																		
D3																		
D2																		
D1																		
D0																		
N A M E	<i>15E</i>	<i>Z</i>	<i>X</i>	<i>W</i>	<i></i>	<i></i>	<i></i>	<i></i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>	<i>G</i>	<i>EN</i>	<i></i>	<i></i>

ALL OUTPUT POLARITIES ARE INVERTED. SO THAT AN "A" CAUSES A "LOW" OUTPUT, A "-" CAUSES A "HIGH" OUTPUT.

SINCE ALL OUTPUTS ARE INVERTED, THE FEEDBACK CIRCUITS FOR THE LATCHES ARE ALSO INVERTED.

THE D02 OUTPUT CREATES THE "ERRR" DISPLAY IF INPUT > (0 TO 9).

TB01205

# AN11 PLD Programmable Retriggerable One-Shot

## Application Note

### Application Specific Products

Author: David Wong

### FEATURES

- Programmable pulse-width/delay
- Maximum 256 clock cycles
- Asynchronous TRIGGER Input
- Active-High and Active-Low outputs
- Asynchronous RESET
- 20-pin package

### THEORY OF OPERATION

The one-shot consists of an FPLS PLS159 and an external clock which may be part of the system in which this one-shot is to work. As shown in Figure 1 and Table 1 the FPLS is configured to have a latch and an eight-bit binary up counter which is presettable by input data to any number less than 256. Since the input data is inverted before it is loaded into the registers, counting from the comple-

ments of the input to FF will give the correct number of counts as counting from the input down to 00.

Pulse-width/delay inputs may be the outputs of another device or switches. When /RESET goes Low, flip-flops are set to all 1's (terms PB and PA). At the rising edge of the next clock, data is latched into the registers (terms LB and LA). When /TRIG goes Low, it is latched into the input latch formed by term # 0, 1, 2 and 13. The output O<sub>1</sub> of the latch goes High and O<sub>2</sub> goes Low which enables the 8-bit counting cycle. The O<sub>1</sub> and /O<sub>1</sub> will maintain their output levels until the end of the counting cycle at which time the counter reaches the count FF, resets the latch by term # 13, and sets O<sub>2</sub> High. At the rising edge of the next clock, terms LA and LB cause data to be loaded again into the registers, and the device is ready for another /TRIG input. The output wave-forms are illustrated in Figure 2.

If the /TRIG pulse-width is longer than the desired pulse-width of the one-shot, the device will react as mentioned above, and at the end of the count cycle new data will be loaded, another count cycle begins while the outputs remain set by the /TRIG input without changing throughout the change-over of one count cycle to another. O<sub>1a</sub>, on the other hand, will go Low for one clock period at the change-over. As long as the /TRIG is Low, O<sub>1a</sub> will continue to pulse Low for one clock period at the change-over of one count cycle to another. The output O<sub>2</sub> will pulse High for one clock cycle at the change-over. Figure 2 illustrates output wave-forms for both cases. The output wave-forms are as illustrated in Figure 2.

The one-shot is implemented by programming the PLS159 as shown in Table 1. The logic representation of the program is shown in Figure 3.

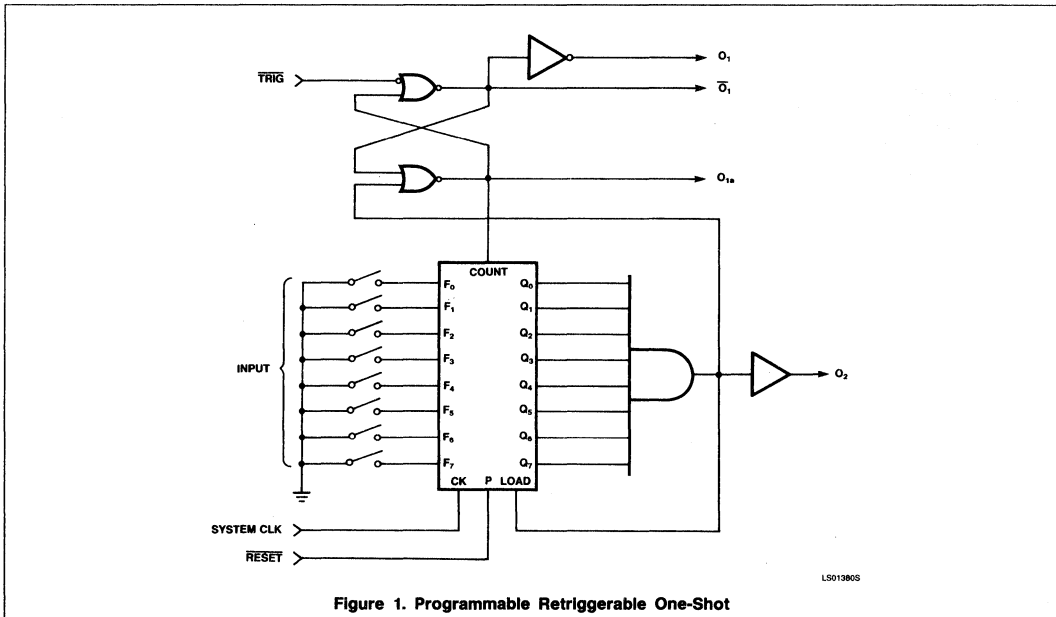


Figure 1. Programmable Retriggerable One-Shot



# PLD Programmable Retriggerable One-Shot

# AN11

Table 1. PLS159 FPLS Program Table

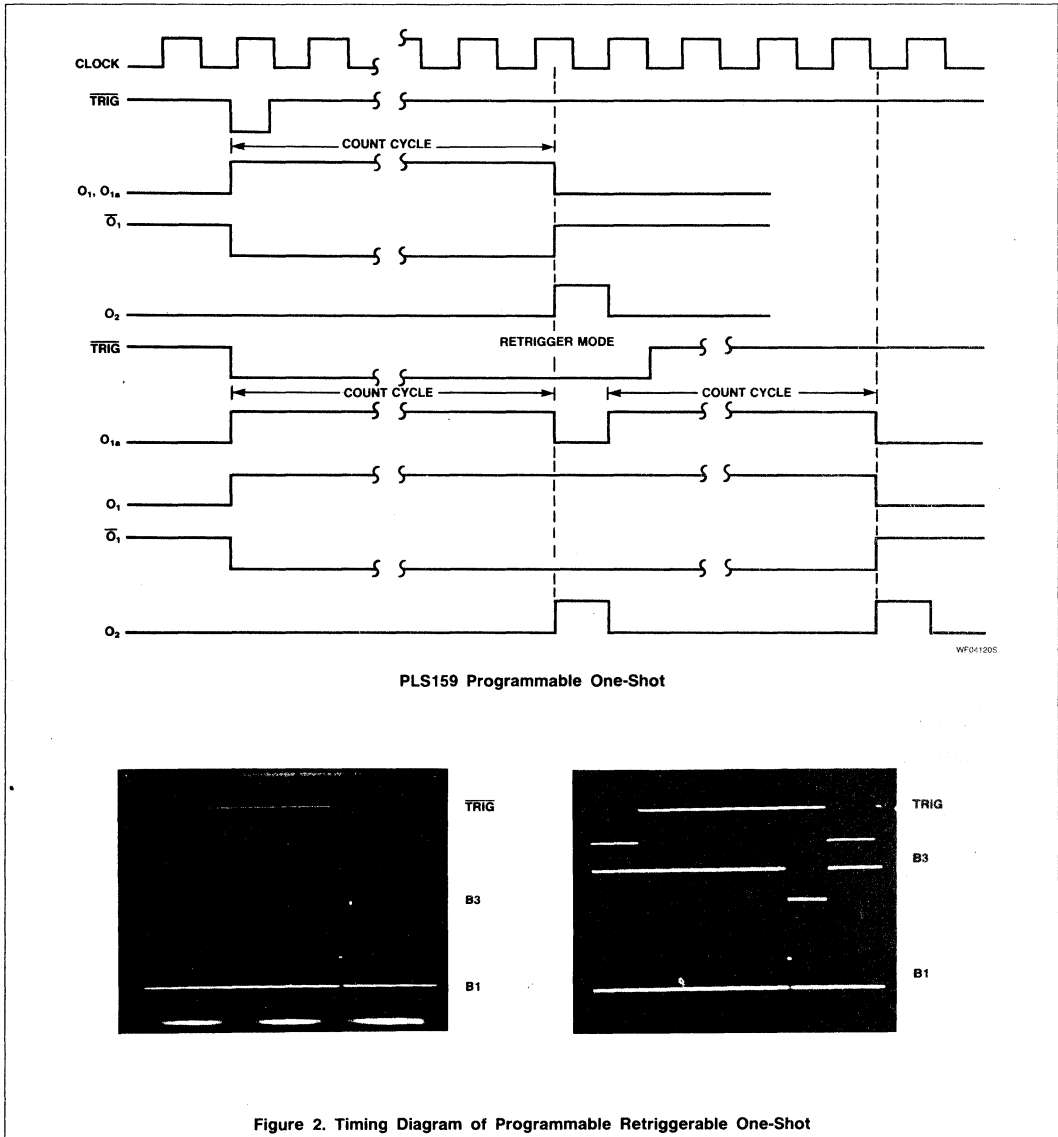
TERMC		PROGRAMMABLE RETRIGGERABLE ONE-SHOT																REMARK	POLARITY				REMARK			
		AND								OR									POLARITY							
		A				B				Q(N)				B(O)					L	L	H	H				
T	C	3	2	1	0	3	2	1	0	7	6	5	4	3	2	1	0	3	2	1	0					
0	-	L	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A	•	•	A					B <sub>0</sub> = B <sub>3</sub>
1	-	-	-	-	-	H	-	-	-	-	-	-	-	-	-	-	-	A	•	•	A					
2	-	-	-	-	-	H	-	-	-	-	-	-	-	-	-	-	-	•	A	•						
3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
4	-	-	H	-	-	H	-	-	-	-	-	-	-	0	•	•	•	•								
5	-	-	H	-	-	H	-	-	-	-	-	-	-	0	•	•	•	•								
6	-	-	H	-	-	H	-	-	-	-	-	-	-	0	•	•	•	•								
7	-	-	H	-	-	H	-	-	-	-	-	-	H	H	H											
8	-	-	H	-	-	H	-	-	-	-	-	-	H	H	H	H										
9	-	-	H	-	-	H	-	-	-	-	-	-	H	H	H	H										
0	-	-	H	-	-	H	-	-	-	-	-	-	H	H	H	H										
11	-	-	H	-	-	H	-	-	-	-	-	-	H	H	H	H										
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-										
13	-	-	-	-	-	-	H	H	H	H	H	H	H	H	H											
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-										
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-										
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-										
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-										
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-										
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-										
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-										
21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-										
22	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-										
23	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-										
24	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-										
25	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-										
26	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-										
27	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-										
28	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-										
29	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-										
30	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-										
31	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-										
2	F <sub>c</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
3	P <sub>e</sub>	-	-	L	-	-	-	-	-	-	-	-	-	-	-	-	-								SET Q <sub>4</sub> TO Q <sub>7</sub> HIGH	
4	R <sub>e</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
5	L <sub>e</sub>	-	-	-	-	-	-	-	-	H	H	H	H	H	H	H								LOAD DATA AT NEXT CLK		
6	P <sub>a</sub>	-	-	L	-	-	-	-	-	-	-	-	-	-	-	-	-								SET Q <sub>6</sub> TO Q <sub>5</sub> HIGH	
7	R <sub>a</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
8	L <sub>a</sub>	-	-	-	-	-	-	-	-	H	H	H	H	H	H	H								LOAD DATA AT NEXT CLK		
9	D <sub>3</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
0	D <sub>2</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
1	D <sub>1</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
2	D <sub>0</sub>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
	PIN	5	4	3	2	9	8	7	6	19	18	17	16	15	14	13	12									
	NAMES	/TRIG	/RESET	O <sub>1a</sub>	O <sub>1</sub>	O <sub>2</sub>	O <sub>3</sub>																			

COMMENTS:  
 This one-shot will load data at the end of the count cycle. If TRIG pulse-width is longer than the count cycle, output B3 will go Low for one clock period and go High again for another count cycle. Outputs B<sub>2</sub> and B<sub>0</sub> stay Low and High respectively until TRIG goes High and count cycle is completed without interruption.

TB004215

# PLD Programmable Retriggerable One-Shot

AN11



# PLD Programmable Retriggerable One-Shot

AN11

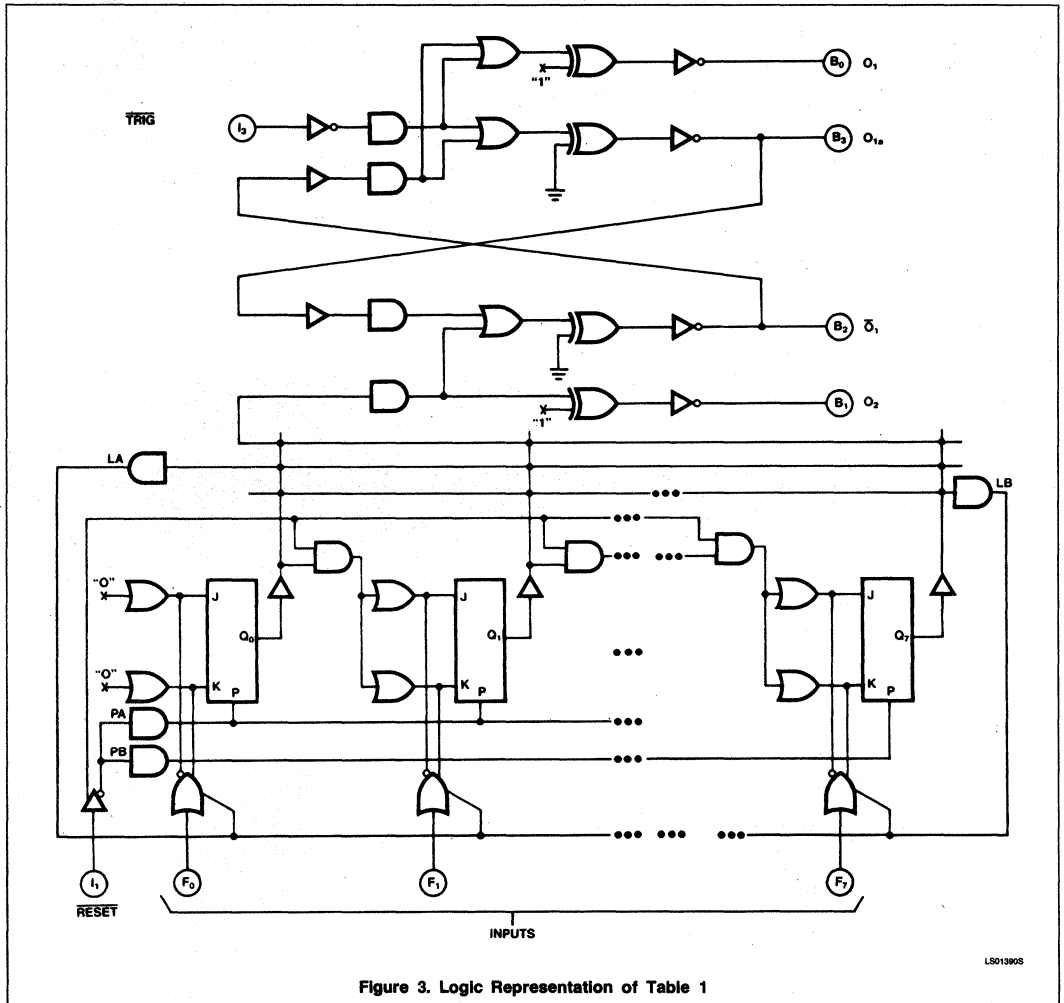


Figure 3. Logic Representation of Table 1

LS013905

## AN12 Low Cost Programmer for PLD 20-Pin Series

### Application Note

#### Application Specific Products

#### INTRODUCTION

The Low Cost Programmer (LCP) is designed to provide the user with a low cost alternative to program Signetics' PLD 20-pin series programmable logic devices. This system is a peripheral device which is to be controlled by a computer running Signetics software, AMAZE, or by a terminal. The LCP is connected to the host computer via the RS232 interface and will accept ASCII codes. The data transfer rate can be either 9600 or 4800 baud. The LCP system clock is 4800 baud. The basic measure of time in the system is in units of 13 $\mu$ s. After every fuse is programmed, the LCP will automatically perform a verification cycle and feed data back to the computer. The system can be set to TEST mode which causes the outputs to cycle through the fusing sequence in an endless loop.

#### THEORY OF OPERATION

A block diagram of the LCP is shown in Figure 1. U1 generates the 9600 baud clock for U3 (UART) and the 4800 baud clock for the rest of the system. When data is transferred into U3, it is put on the system bus. U1 generates IDAV which enables the rest of the system to react to the input. Since each device can decode the input instruction, very few controls are necessary.

Figure 2 shows the system instruction set. Each consists of seven bits with the upper 3 bits being the instruction code and the lower 4 bits being data or finer resolution of the instruction code. Figures 2 and 3 of Appendix 1 show the fuse address tables of PLS153 and PLS159, respectively. Note that the Variable Addresses have seven bits (3 upper bits and 4 lower bits), whereas the Term Addresses have six bits (2 upper bits and 4 lower bits). Therefore, it takes up to four nibbles to define a fuse address. Once an address is defined, the Variable Address generator will increment automatically to generate the subsequent fuse addresses for the same Term Address. Loading a Term Address in the Term Address generator will always reset the Variable Address generator. Therefore, the Term Address should be entered first and then the Variable Address. The last entry is the fusing data in a 4-bit nibble. The system will automatically increment the Variable Address generator to program four fuses. Once the 4-bit fusing data is loaded, the system goes into an internal mode until

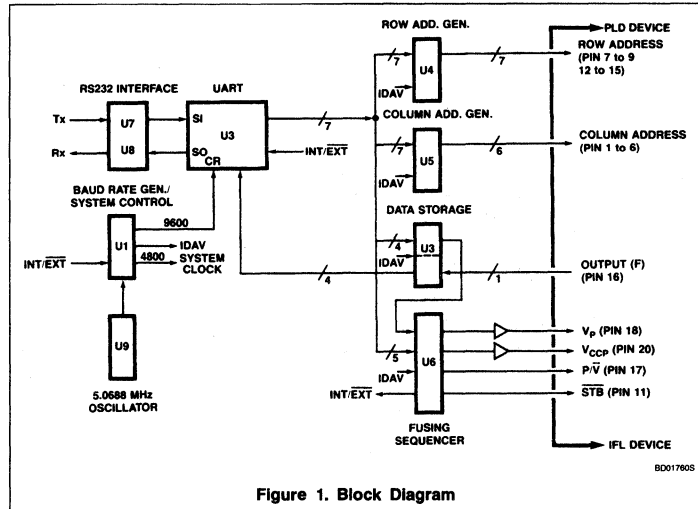


Figure 1. Block Diagram

C2 (D6)	C1 (D5)	C0 (D4)	D3	D2	D1	D0	FUNCTION
0	0	0	X	X	X	X	System internal use
0	0	1	X	X	X	X	System internal use
0	1	0	0	0	0	0	System reset
0	1	0	0	0	0	1	Set system ON-LINE
0	1	0	0	0	1	0	Set system OFF-LINE
0	1	1					4-bit data
1	0	0					4-bit data
1	0	1					2-bit data
1	1	0					4-bit data
1	1	1					3-bit data

**NOTE:**  
The upper and lower VARIABLE ADDRESSES will be reset to zero if upper or lower TERM ADDRESS is loaded into the LCP.

Figure 2. LCP Instruction Set

the fusing cycle for the four fuses are completed.

The following illustrates the sequence of programming operation:

1. Switch LCP to ON-LINE
2. Input 010 0000 - reset LCP
3. Input 010 0001 - put LCP "ON-LINE"

# Low Cost Programmer for PLD 20-Pin Series

# AN12

4. Input 100 --- load 4-bit lower TERM ADDRESS
5. Input 101 00--- load 2-bit upper TERM ADDRESS
6. Input 110--- -- load 4-bit lower VARIABLE ADDRESS
7. Input 111 0--- -- load 3-bit upper VARIABLE ADDRESS
8. Input 011--- -- load 4-bit fusing data, a "1" causes a fuse to be blown
9. U6, the fusing sequencer, causes the system to go into INTERN mode
10. Fusing sequence starts. See Figures 3 and 4
11. Fusing cycle completed. Input next address:
  - a. If next TERM ADDRESS is different from the previous one, go to (4)
  - b. If next address is in sequence with the previous VARIABLE ADDRESS and in the same TERM, go to (6)
  - c. The Variable Address generator will always reset to 0 if TERM ADDRESS is loaded.

Figure 3 shows the sequence of fusing operation while the system is in internal mode. Figure 4 shows the timing diagram and voltage waveform of the fusing sequence of four consecutive fuses.

### Analog Drivers:

**V<sub>p</sub>** — This signal is current limited to 325mA. It is 1.25V to 1.75V when not fusing (idle). During the fusing cycle it takes on a value of 14.25V to 14.75V for a short time. The leading edge is slow rate limited to about 10μs.

**V<sub>CCP</sub>** — This signal is the programming power supply. It can source up to 1A. It has a value of 1.1V to 1.3V when not fusing (idle).

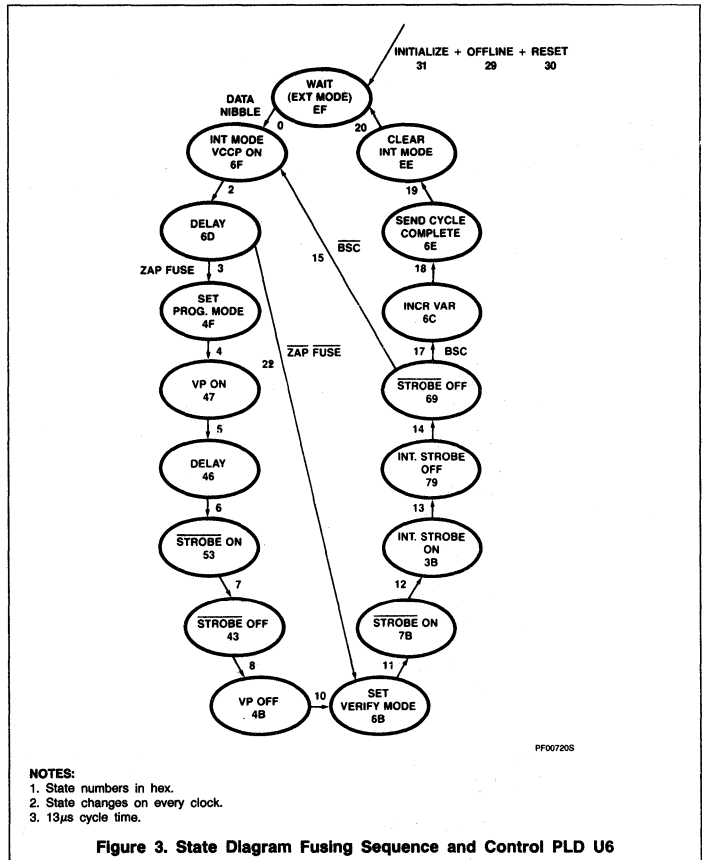


Figure 3. State Diagram Fusing Sequence and Control PLD U6

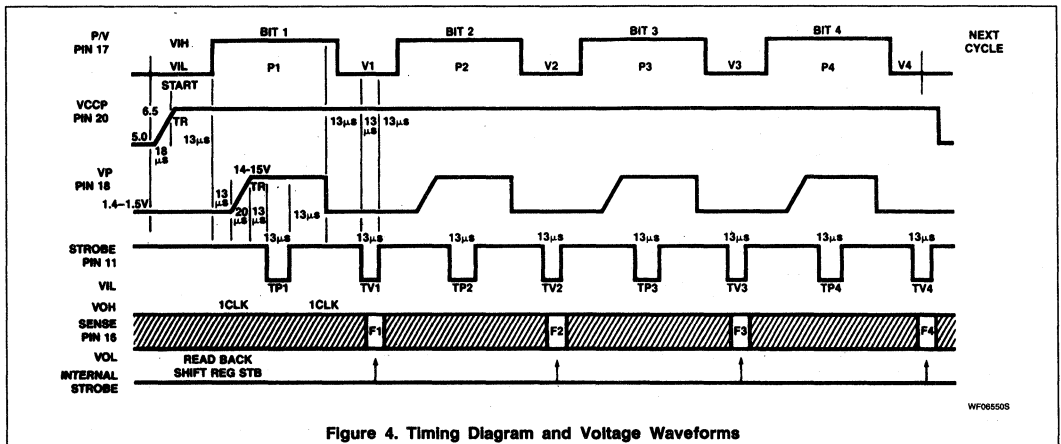


Figure 4. Timing Diagram and Voltage Waveforms

# Low Cost Programmer for PLD 20-Pin Series

# AN12

During the fusing cycle it assumes a value of 8.25V to 8.75V for the duration of the entire cycle (four fusing pulses—See Figure 4).

**V<sub>p</sub> Driver Circuitry** — The V<sub>p</sub> signal is toggled by the Fusing Sequencer, U6, Pin 15. This TTL signal is buffered up by the switching transistor, Q3, and its associated resistors, R6 and R9. Transistor Q4 inverts the signal to achieve the correct phase relationship. It also performs the following circuit functions:

1. It clamps the base of the Darlington transistor, Q5, to near ground through the R10 resistor.
2. Transistor Q4 discharges the slew rate time constant capacitor, C1, through the resistor, R10, and shunts the current from the current source, VR2, to near ground. This prepares the circuit for the next controlled slew rate edge at the beginning of the next fuse blowing cycle.
3. It provides a current sinking path to pull current out of the fusing socket enable line, thus creating a rapidly falling V<sub>CCP</sub> trailing edge. The diode, D2, provides sourcing isolation while the resistor, R11, limits inrush currents. A ferrite bead on the R11 lead slows down this path. This prevents large currents from flowing through transistor Q4 at the low level switching transition time.

The voltage regulator, VR1, forms the 325mA current limit circuit in conjunction with the voltage dropping resistor, R8.

The controlled slew rate of the leading edge and the final level of the enable high output voltage V<sub>p</sub> are generated in the following manner:

1. A 10mA current source is formed by the voltage regulator, VR2, and the dropping resistor, R7.
2. This current is shunted to ground through Q4 when the enable pulse is low.
3. At the instant when Q4 turns off, the 10mA current is diverted into the slew rate control capacitor, C1. The capacitor, C1, will continue to charge linearly until the voltage across it reaches the value of the zener diode stack D1A and D1B.
4. The current is then shunted down the zener diodes until Q4 is turned back on.
5. The base of the Darlington transistor, Q5, follows the voltage across the capacitor, C1, and draws very little current due to its very high current gain. Since the Q5 transistor is configured as an emitter follower, the voltage seen on the fusing socket enable line will be about 1V lower than the base of transistor Q5. Resistor R12 stabilizes the value of the V<sub>BE</sub> on transistor Q5 by pulling a small amount of current early in the ramp.

**V<sub>CCP</sub> Driver Circuitry** — V<sub>CCP</sub> is toggled by the internal/external signal from the Fusing Sequencer, U6. Again, the signal is buffered and inverted twice by transistors Q1 and Q2 along with resistors R4 and R3.

**Low Voltage (Idle)**

Resistor R2 is shorted to ground through transistor Q2 in order to generate the 1.1V to 1.3V low output level. This is an intrinsic value generated by the internal reference of the voltage regulator.

**High Voltage (V<sub>CCP</sub>)**

When transistor Q2 is turned off, the output voltage ramps to the 8.25V to 8.75V level. This value is determined by the ratio of resistors R1 and R2 in conjunction with the actual value of the voltage regulator's internal reference.

**Power Supply Consideration**

1. The Low Cost Programmer circuitry draws about 1A.
2. The Enable and V<sub>CCP</sub> currents may reach about 1.4A.
3. The Enable signal is about 15V and needs at least 5V of head room.
4. The RS232 drivers require ± 12V at about 60mA.

In a bridge circuit, a rule of thumb says that the transformer's current rating should be from 1.2 to 1.6 times the load current. Using this rule, the secondary of the transformer that supplies V<sub>CC</sub> and ± 12V should be rated at 18V center tapped at about 1.6A.

The transformer that supplies the Enable and V<sub>CCP</sub> circuits should have an 18V secondary at about 2.2A.

**PROGRAMMING THE LCP**

The addresses and fusing data can be generated by the host computer using AMAZE or a terminal nibble by nibble. The fuse address of each PLD 20-pin device is available in the Signetics PLD data manual. Copies of the fuse address tables of PLS153 and PLS159 are included in Appendix A Tables A-1 and A-2. These fuse addresses, together with the 3-bit instruction code, form a 7-bit instruction which can be represented by a single ASCII character. Table 1 is a table of ASCII characters.

The following example illustrates the conversion of program table entries into fuse addresses and fusing data. Table 2 shows a simple entry in an PLS153 program table. To keep things simple, only four entries are made. The first entry is term 3, variable 17. The fuse of the non-inverting buffer is to remain intact, whereas the fuse to be blown is the inverting buffer, 17. For this entry, from Table A-1 of Appendix A, the Term Address is 03hex or 00 0011b, the Variable Addresses of the two buffers are 0Ehex or 000 1110b, and 0Fhex or 000 1111. The fusing data is 01. The second entry is Term 3, 16. The fuse of the inverting buffer is to be kept intact while that of the non-inverting buffer will be blown. The Term Address is the same as the previ-

**Table 1. ASCII Character Table**

		0000	0001	0010	0011	0100	0101	0110	0111
		0	1	2	3	4	5	6	7
0000	0	NUL	DLE	SP	0	●	P	·	p
0001	1	SOH	DC1	!	1	A	Q	a	q
0010	2	STX	DC2	"	2	B	R	b	r
0011	3	ETX	DC3	#	3	C	S	c	s
0100	4	EOT	DC4	\$	4	D	T	d	t
0101	5	ENQ	NAK	%	5	E	U	e	u
0110	6	ACK	SYN	&	6	F	V	f	v
0111	7	BEL	ETB	'	7	G	W	g	w
1000	8	BS	CAN	(	8	H	X	h	x
1001	9	HT	EM	)	9	I	Y	i	y
1010	A	LF	SUB	*	:	J	Z	j	z
1011	B	VT	ESC	+	;	K	[	k	{
1100	C	FF	FS	,	<	L	^	l	:
1101	D	CR	GS	—	=	M	]	m	}
1110	E	SO	RS	.	>	N		n	~
1111	F	SI	US	/	?	O	÷	o	DEL



# Low Cost Programmer for PLD 20-Pin Series

AN12

Table 2. FPLA Program Table

CODE NO.		AND																	POLARITY																	OR																
T E R M	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	REMARKS	9	8	7	6	5	4	3	2	1	0	REMARKS																						
0																																																				
1																																																				
2																																																				
3	H	L	-	-																																																
4																																																				
5																																																				
6																																																				
7																																																				
8																																																				
9																																																				
10																																																				
11																																																				
12																																																				
13																																																				
14																																																				
15																																																				
16																																																				
17																																																				
18																																																				
19																																																				
20																																																				
21																																																				
22																																																				
23																																																				
24																																																				
25																																																				
26																																																				
27																																																				
28																																																				
29																																																				
30																																																				
31																																																				
D9																																																				
D8																																																				
D7																																																				
D6																																																				
D5																																																				
D4																																																				
D3																																																				
D2																																																				
D1																																																				
D0																																																				
PN	8	7	6	5	4	3	2	1	19	18	17	16	15	14	13	12	11	9																																		
REMARKS																																																				

PLS153

TB016505

# Low Cost Programmer for PLD 20-Pin Series

# AN12

ous entry; the Variable Addresses of the two buffers are 0Chex or 000 1100b and 0Dhex or 000 1101b. The fusing data is 10. The four fusing data can be combined into one nibble (0110) since the two entries are in sequence. Only the first address has to be loaded (Term Address 03hex and Variable Address 0F). The next two entries are in sequence with the first two, therefore only fusing data nibble 1111 has to be loaded.

The instruction and data nibbles can be loaded into the LCP in ASCII characters (see Table 1).

## LCP CHECK OUT AND CALIBRATION

This section deals with the check out and calibration of the Low Cost Programmer.

The voltage regulator, VR4, dissipates a lot of heat. It should be mounted on a metal chassis with an isolation mount.

Notice that the fusing socket, P1, has three columns of pads. Use the left set if the socket is to be mounted on the component side of the PCB. Use the right set if the socket is to be placed on the blank side of the PCB. The LEDs may also be mounted on the blank side, allowing for very easy panel mounting of the finished LCP.

### Power Supplies

- +10V<sub>DC</sub> to +14V<sub>DC</sub> at the positive side of C4.
- 10V<sub>DC</sub> to -14V<sub>DC</sub> at the negative side of C3.
- +4.75V<sub>DC</sub> to +5.25V<sub>DC</sub> at the +5V<sub>DC</sub> input pad of the PCB.
- +20V<sub>DC</sub> to +26V<sub>DC</sub> at the positive side of C2.

Turn off the LCP and insert ICs U1 through U5, U7 and U8. Do not insert U6 yet.

Turn on the LCP and repeat steps 1 through 4 above.

### Driver Calibration

The Fusing Sequencer, U6, must not be in the LCP for the following steps:

- Measure 1.25V<sub>DC</sub> to 1.75V<sub>DC</sub> at Pin 18 of P1. (The fusing socket.)
- Place a clip lead from the U6 Pin 15 end of resistor R6 to V<sub>CC</sub> (+5V<sub>DC</sub>).
- Measure 14.0V<sub>DC</sub> to 15.0V<sub>DC</sub> at Pin 18 of P1.
- Momentarily place a 75Ω 3W resistor from Pin 18 of P1 to ground and observe the voltage to still read 14.0V<sub>DC</sub> to 15.0V<sub>DC</sub>.
- Disconnect the clip lead between R6 and V<sub>CC</sub>.

- Measure the voltage at Pin 18 of P1 to once again be 1.25 to 1.75V<sub>DC</sub>.
- Place a jumper from the U6 Pin 19 side of resistor R4 to ground.
- Measure the voltage at Pin 20 of P1. It should be 1.10V<sub>DC</sub> to 1.30V<sub>DC</sub>.
- Place a jumper from the U6 Pin 19 side of resistor R4 to V<sub>CC</sub>.
- The voltage should be from 8.25V<sub>DC</sub> to 8.75V<sub>DC</sub> at Pin 20 of P1.
- Place a 30Ω 3W load resistor momentarily from Pin 20 of P1 to ground and measure the voltage. It should remain within the 8.25V<sub>DC</sub> to 8.75V<sub>DC</sub> range. The value of R2 may have to be selected in order to give the correct voltage reading.
- Reconnect the jumper from R4 to ground.
- The voltage at Pin 20 of P1 should be 1.10V<sub>DC</sub> to 1.30V<sub>DC</sub>.
- Turn off power and insert U6 into its socket.

### Baud Rate Generation

- Using an oscilloscope, look at the waveform on Pin 17 of the Baud Rate Generator, U1. This signal should swing from < 450mV to > 2.4V and have about a 50% duty cycle.
- The frequency at this pin should be 153.6kHz. This results in a period of 6.51μs.
- Repeat Steps 1 and 2 for Pin 18 of U1. The frequency should be 76.8kHz, resulting in a period of 13.0μs.

### Control Functions

Connect an ASCII RS232 terminal to the computer connector, C1, on the LCP. The set up should be:

- 9600 baud.
- 8 bits.
- 1 Stop bit.
- No parity.
- Terminal Tx to Pin 3 of the LCP C1 connector.
- Terminal Rx to Pin 2 of the LCP C1 connector.

The following steps send ASCII characters to the LCP, which it should interpret as instructions. It will not echo the characters sent, but it should perform the instructions. Refer to Figure 5 for the ASCII equivalent instruction characters.

- Turn on the LCP.
- Set the ON-LINE/OFF-LINE switch to ON-LINE.
- Type " .
- Type !. The ON-LINE LED should illuminate. If the ON-LINE LED does not light at this point, check for the correct inter-

facing of Pins 2 and 3 on the RS232 terminal connector.

- Type ". The ON-LINE LED should go back off.

### TERM ADDRESS GENERATION

The purpose of this section is to test the ability of the LCP to uniquely load the bits of the term address. This is done by sending ASCII characters to the LCP and then checking the term address pins of P1.

Send the following lines of characters and check the P1 pins after each.

CHAR	1	2	3	4	5	6
!	On-Line — Address at Previous State					
SO	H	H	H	H	H	H
space	L	L	L	L	L	L
A	L	L	L	L	L	L
B	L	L	L	L	H	H
D	L	L	L	H	L	L
H	L	L	H	L	L	L
@Q	L	H	L	L	L	L
R	H	L	L	L	L	L
SO	H	H	H	H	H	H
N	H	H	H	H	H	L
M	H	H	H	H	L	H
K	H	H	H	L	H	H
G	H	H	L	H	H	H
OR	H	L	H	H	H	H
Q	L	H	H	H	H	H
S	H	H	H	H	H	H
@ P	L	L	L	L	L	L
	Off-Line — Address Not Altered					

### VARIABLE ADDRESS GENERATION

The purpose of this section is to test the ability of the LCP to uniquely load and increment the bits of the Variable Address. This is done by sending the appropriate ASCII characters to the LCP and then checking the Term Address pins of P1.

Send each line of characters and check the specified P1 pins after each.





# Low Cost Programmer for PLD 20-Pin Series

# AN12

CHAR	7	8	9	12	13	14	15
l	On-Line — Address Same as Previous						
wo	H	H	H	H	H	H	H
space	L	L	L	L	L	L	L
a	L	L	L	L	L	L	H
b	L	L	L	L	L	H	L
d	L	L	L	L	H	L	L
h	L	L	L	H	L	L	L
space q	L	L	H	L	L	L	L
r	L	H	L	L	L	L	L
t	H	L	L	L	L	L	L
ow	H	H	H	H	H	H	H
n	H	H	H	H	H	H	L
m	H	H	H	H	H	L	H
k	H	H	H	L	L	H	H
g	H	H	H	L	H	H	H
ov	H	H	L	H	H	H	H
u	H	L	H	H	H	H	H
s	L	H	H	H	H	H	H
w	H	H	H	H	H	H	H
A	L	L	L	L	L	L	L
owQ	L	L	L	L	L	L	L
1	L	L	L	L	H	L	L
1	L	L	L	H	L	L	L
2	L	L	L	H	H	L	L
2	L	L	H	L	L	L	L
l2	L	H	L	L	L	L	L
ls2	H	L	L	L	L	L	L
	Off-Line — Address Not Altered						

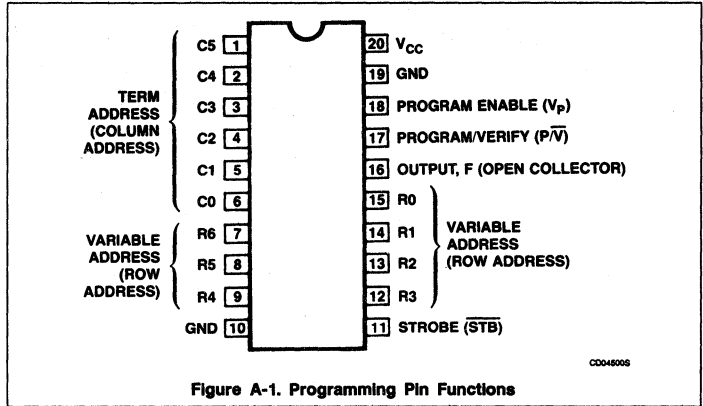


Figure A-1. Programming Pin Functions

Type on the terminal keyboard. The computer should behave normally — usually with an echo.

This completes the LCP calibration and checkout.

### APPENDIX A

#### HOW TO PROGRAM PLD 20-PIN SERIES DEVICES

An PLD device is basically a group of logic functions interconnected to each other by fuses which can be blown to open a circuit, or left intact to make a connection. Each fuse can be identified by its own unique address which consists of a 7-bit Variable Address (ROW ADDRESS) and a 6-bit Term Address (COLUMN ADDRESS). The programming pin function is the same throughout the 20-pin series see Figure A-1). Tables A-1 and A-2 are fuse address tables of the PLS153 and PLS159, respectively.

The programming cycle can be separated into four stages:

#### Set-Up

1. Set Program Enable ( $V_p$ ) to GND

2. Apply Column and Row Address to pins 1-6 and 7-9, 11-15
3. Set STROBE (STB) to  $V_{IH}$
4. Set PROGRAM/VERIFY (P/V) to  $V_{IL}$
5. Wait 1-50 $\mu$ s
6. Set  $V_{CCP}$  to 8.5V

#### To Blow a Fuse

1. Wait 1-10 $\mu$ s
2. Set P/V to  $V_{IH}$
3. Wait 1-50 $\mu$ s
4. Set  $V_p$  to 14.5V
5. Wait 1-5 $\mu$ s
6. Pulse STB to  $V_{IL}$  for 10 $\mu$ s
7. Wait 1-5 $\mu$ s
8. Set  $V_p$  to GND
9. Wait 1-5 $\mu$ s
10. Set P/V to  $V_{IL}$

#### To Verify

1. Wait 1-5 $\mu$ s
2. Pulse STB to  $V_{IL}$  for 1-10 $\mu$ s
3. While STB is LOW, OUTPUT (F) will be  $V_{IL}$  for a fuse intact and 2-8.75V for a blown fuse (note: F is an open-collector, a pull-up circuit may be needed)

#### Go To Set-Up for Next Fuse

# Low Cost Programmer for PLD 20-Pin Series

# AN12

**Table A-1. Program Cycle Row/Column Fuse Addressing**

Variable Select Table <sup>1</sup>				PLS153				Term Select Table <sup>2</sup>			
ROW HEX ADDRESS		SELECTED VARIABLE	ROW HEX ADDRESS		SELECTED VARIABLE	COLUMN HEX ADDRESS		SELECTED PRODUCT TERM			
I <sub>6</sub> I <sub>7</sub> B <sub>0</sub>	B <sub>2</sub> B <sub>3</sub> B <sub>4</sub> B <sub>5</sub>		I <sub>6</sub> I <sub>7</sub> B <sub>0</sub>	B <sub>2</sub> B <sub>3</sub> B <sub>4</sub> B <sub>5</sub>		I <sub>0</sub> I <sub>1</sub>	I <sub>2</sub> I <sub>3</sub> I <sub>4</sub> I <sub>5</sub>				
0	0	AND Array	I <sub>0</sub>	2	4	Empty Address Space	0	0	0		
0	1		I <sub>0</sub>	2	4		0	1	1		
0	2		I <sub>1</sub>	2	4		0	2	2		
0	3		I <sub>1</sub>	2	4		0	3	3		
0	4		I <sub>2</sub>	2	4		0	4	4		
0	5		I <sub>2</sub>	2	4		0	5	5		
0	6		I <sub>3</sub>	2	4		0	6	6		
0	7		I <sub>3</sub>	2	4		0	7	7		
0	8		I <sub>4</sub>	2	4		0	8	8		
0	9		I <sub>4</sub>	2	4		0	9	9		
0	A		I <sub>5</sub>	3	0	OR Array	0	A	10		
0	B		I <sub>5</sub>	3	1		9	B	11		
0	C		I <sub>6</sub>	3	2		8	C	12		
0	D		I <sub>6</sub>	3	3		7	D	13		
0	E		I <sub>7</sub>	3	4		6	E	14		
0	F		I <sub>7</sub>	3	5		5	F	15		
1	0		I <sub>7</sub>	3	6	4	Logic Terms	1	0	16	
1	1		I <sub>7</sub>	3	7	3		1	1	17	
1	1		B <sub>9</sub>	3	8	2		1	2	18	
1	1		B <sub>9</sub>	3	9	1		1	3	19	
1	2	B <sub>9</sub>	3	9	0	1		4	20		
1	3	B <sub>8</sub>	3	9	0	1		5	21		
1	4	B <sub>8</sub>	3	A	Polarity Enable			1	6	22	
1	5	B <sub>8</sub>	3	A	Polarity Enable			1	7	23	
1	6	B <sub>7</sub>	3	B	Empty Address Space	1		8	24		
1	7	B <sub>7</sub>	3	B		1		9	25		
1	8	B <sub>6</sub>	3	B		1	A	26			
1	9	B <sub>6</sub>	3	B		1	B	27			
1	A	B <sub>5</sub>	3	B		1	C	28			
1	A	B <sub>5</sub>	3	B		1	D	29			
1	B	B <sub>4</sub>	3	B		1	E	30			
1	B	B <sub>4</sub>	3	B		1	F	31			
1	C	B <sub>3</sub>	3	B		2	0	Control Terms	2	0	
1	D	B <sub>3</sub>	3	B		2	1		1	1	
1	E	B <sub>2</sub>	3	B	2	2	2		2		
1	E	B <sub>2</sub>	3	B	2	3	3		3		
1	F	B <sub>2</sub>	3	B	2	4	4		4		
1	F	B <sub>2</sub>	3	B	2	5	5		5		
1	0	B <sub>1</sub>	3	B	2	6	6		6		
1	1	B <sub>1</sub>	3	B	2	7	7		7		
1	1	B <sub>1</sub>	3	B	2	8	8		8		
1	2	B <sub>0</sub>	3	B	2	9	9		9		
1	2	B <sub>0</sub>	3	B	2	A	Polarity Terms	2	A	0	
1	3	B <sub>0</sub>	3	B	2	B		1	1		
1	3	B <sub>0</sub>	3	B	2	C		2	2		
1	3	B <sub>0</sub>	3	B	2	D		3	3		
1	3	B <sub>0</sub>	3	B	2	E		4	4		
1	3	B <sub>0</sub>	3	B	2	F		5	5		
1	3	B <sub>0</sub>	3	B	3	0		6	6		
1	3	B <sub>0</sub>	3	B	3	1		1	1		
1	3	B <sub>0</sub>	3	B	3	2		8	8		
1	3	B <sub>0</sub>	3	B	3	3		9	9		

**NOTES:**

- 1. A row address identifies a particular variable coupled to all product terms.
- 2. With a variable selected by the row address the column address further selects a coupling fuse for each term.

# Low Cost Programmer for PLD 20-Pin Series

AN12

Table A-2. Program Cycle Row/Column Fuse Addressing (Continued)

Variable Select Table <sup>1</sup>				PLS159				Term Select Table <sup>2</sup>				
ROW HEX ADDRESS		SELECTED VARIABLE		ROW HEX ADDRESS		SELECTED VARIABLE		COLUMN HEX ADDRESS		SELECTED TERM		
B <sub>1</sub> B <sub>2</sub> B <sub>3</sub>	F <sub>0</sub> F <sub>1</sub> F <sub>2</sub> F <sub>3</sub>			B <sub>1</sub> B <sub>2</sub> B <sub>3</sub>	F <sub>0</sub> F <sub>1</sub> F <sub>2</sub> F <sub>3</sub>			CLKI <sub>0</sub>	I <sub>1</sub> I <sub>2</sub> I <sub>3</sub> B <sub>0</sub>			
0	0	AND Array	B <sub>3</sub>	5	0	Flip-flop Control Terms	F <sub>7</sub>	0	0	Transition Term	0	
0	1		B <sub>3</sub>	5	1		F <sub>7</sub>	J	0		1	1
0	2		B <sub>2</sub>	5	2		F <sub>6</sub>	J	0		2	2
0	3		B <sub>2</sub>	5	3		F <sub>6</sub>	K	0		3	3
0	4		B <sub>1</sub>	5	4		F <sub>5</sub>	J	0		4	4
0	5		B <sub>1</sub>	5	5		F <sub>5</sub>	K	0		5	5
0	6		B <sub>0</sub>	5	6		F <sub>4</sub>	J	0		6	6
0	7		B <sub>0</sub>	5	7		F <sub>4</sub>	K	0		7	7
0	8		I <sub>3</sub>	5	8		F <sub>3</sub>	J	0		8	8
0	9		I <sub>3</sub>	5	9		F <sub>3</sub>	K	0		9	9
0	A		I <sub>2</sub>	5	A		F <sub>2</sub>	J	0		A	10
0	B		I <sub>2</sub>	5	B		F <sub>2</sub>	K	0		B	11
0	C		I <sub>1</sub>	5	C		F <sub>1</sub>	J	0		C	12
0	D		I <sub>1</sub>	5	D		F <sub>1</sub>	K	0		D	13
0	E		I <sub>0</sub>	5	E		F <sub>0</sub>	J	0		E	14
0	F		I <sub>0</sub>	5	F		F <sub>0</sub>	K	0		F	15
2	0	F <sub>0</sub>	6	0	J-K/D Option	F <sub>7</sub>	1	0	16			
2	1	F <sub>0</sub>	6	1		F <sub>6</sub>	1	1	17			
2	2	F <sub>1</sub>	6	2		F <sub>5</sub>	1	2	18			
2	3	F <sub>1</sub>	6	3		F <sub>4</sub>	1	3	19			
2	4	F <sub>2</sub>	6	4		F <sub>3</sub>	1	4	20			
2	5	F <sub>2</sub>	6	5		F <sub>2</sub>	1	5	21			
2	6	F <sub>3</sub>	6	6		F <sub>1</sub>	1	6	22			
2	7	F <sub>3</sub>	6	7	F <sub>0</sub>	1	7	23				
2	8	F <sub>4</sub>	7	1	OR Array	B <sub>3</sub>	1	8	24			
2	9	F <sub>4</sub>	7	2		B <sub>2</sub>	1	9	25			
2	A	F <sub>5</sub>	7	3		B <sub>1</sub>	1	A	26			
2	B	F <sub>5</sub>	7	4		B <sub>0</sub>	1	B	27			
2	C	F <sub>6</sub>	7	5	C-Array	C <sub>N</sub>	1	C	28			
2	D	F <sub>6</sub>	7	6	OE	EA	1	D	29			
2	E	F <sub>7</sub>	7	7	Polarity Terms	B <sub>0</sub>	1	E	30			
2	F	F <sub>7</sub>	7	8		B <sub>1</sub>	1	F	31			
2	F	F <sub>7</sub>	7	D		B <sub>2</sub>	2	0	PA			
2	F	F <sub>7</sub>	7	F		B <sub>3</sub>	2	1	PB			
4	F	C <sub>N</sub>	7	5	OE	EA	2	2	RA			
				6	OE	EA	2	3	RB			
				7	A	B <sub>0</sub>	2	4	LA			
				8	B <sub>1</sub>	B <sub>1</sub>	2	5	LB			
				D	B <sub>2</sub>	B <sub>2</sub>	2	6	FC			
				F	B <sub>3</sub>	B <sub>3</sub>	2	7				
							2	8	EA			
							2	9	EB			
							2	C	A EA EB			
							2	A				
							2	D	Polarity			
							2	E	D <sub>0</sub> D <sub>1</sub> D <sub>2</sub> D <sub>3</sub>			
							2	F				
							3	0				
							3	1				
							2	0	Test Col. 2			
							3	2	Test Col. 1			

**NOTES:**

1. A row address identifies a particular variable coupled to all product terms.
2. With a variable selected by the row address the column address further selects a coupling fuse for each term.

# Low Cost Programmer for PLD 20-Pin Series

AN12

## APPENDIX B

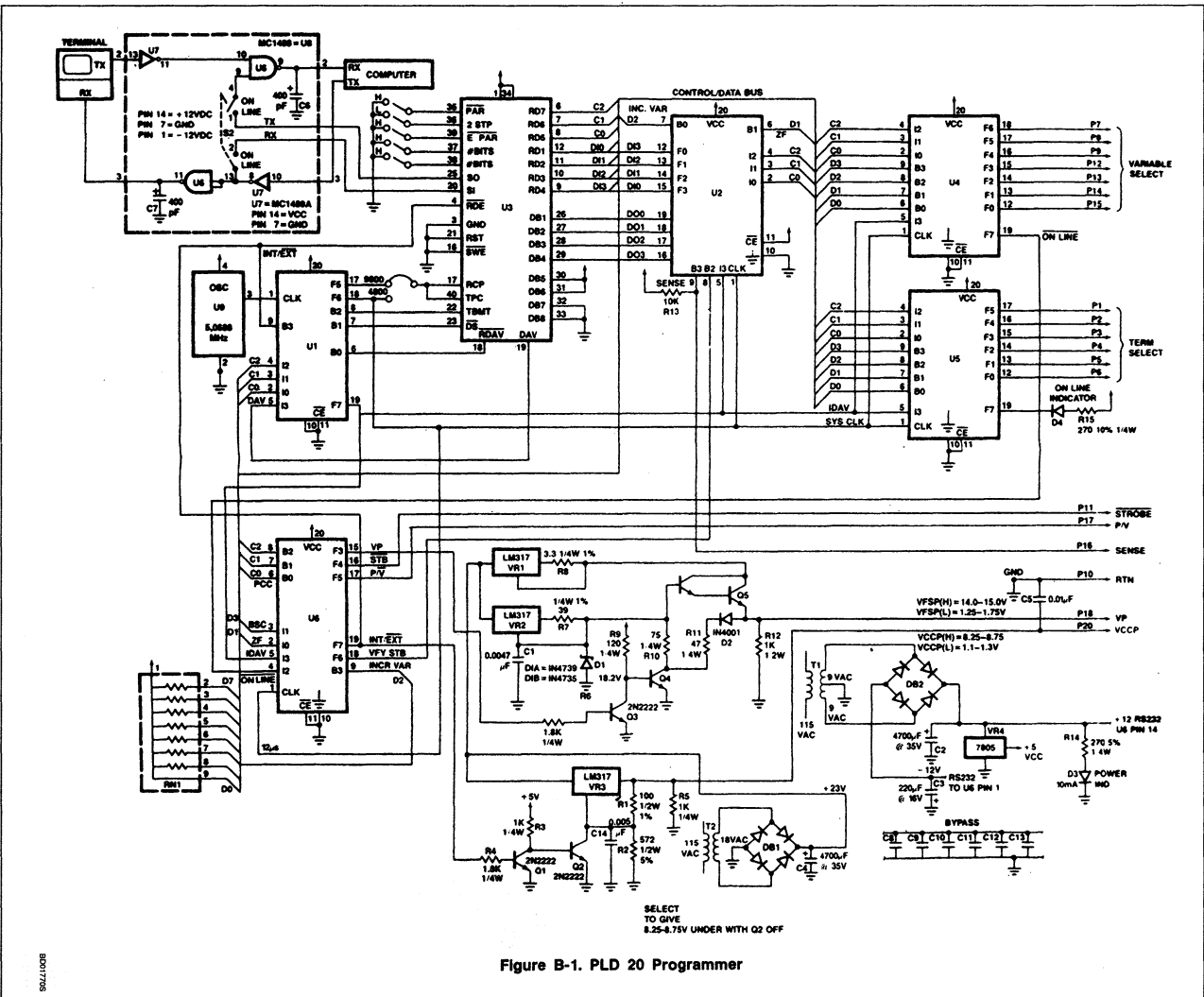


Figure B-1. PLD 20 Programmer

801778S

# Low Cost Programmer for PLD 20-Pin Series

AN12

Table B-1. Baud Rate Generation and UART Control PLD U1 Revision "B" Adds Test Mode

CODE NO.		FF MODE																				E <sub>B</sub>		E <sub>A</sub>		POLARITY							
U1		.																				.	.	.	.	H	H	L	L	L	L		
T E R M	C	AND										REMARKS	(OR)										REMARKS										
		B(I)					Q(P)						Q(N)					B(O)															
		3	2	1	0	3	2	1	0	7	6	5	4	3	2	1	0	3	2	1	0	7	6	5	4	3	2	1	0				
0	.	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	.	.	.	.	.	.	.	.	.	.	.					
1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	.	.	.	.	.	.	.	.	.	.	.					
2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	.	.	.	.	.	.	.	.	.	.	.					
3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	.	.	.	.	.	.	.	.	.	.	.					
4	-	-	-	-	-	-	-	-	-	-	-	L	-	H	H	H	H	.	.	.	.	.	.	.	.	.	.	.					
5	-	-	-	-	-	-	-	-	-	-	H	-	-	H	H	H	H	.	.	.	.	.	.	.	.	.	.	.					
6	A	-	-	-	-	-	-	-	-	-	H	H	L	L	L	L	.	.	.	.	.	.	.	.	.	.	.	.					
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	.	.	.	.	.	.	.					
8	-	H	-	H	-	L	H	-	-	-	H	H	H	L	L	L	L	.	.	.	.	.	.	.	.	.	.	.					
9	-	-	-	-	L	-	-	-	L	H	H	H	L	L	L	L	.	.	.	.	.	.	.	.	.	.	.	.					
10	-	-	-	-	L	-	-	-	L	-	-	-	-	-	-	-	.	.	.	.	.	.	.	.	.	.	.						
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	.	.	.	.	.	.	.						
12	-	-	-	-	L	H	-	-	-	H	H	H	L	L	L	L	.	.	.	.	.	.	.	.	.	.	.						
13	-	-	-	-	H	H	-	-	-	L	H	H	H	L	L	L	.	.	.	.	.	.	.	.	.	.	.						
14	-	-	-	-	H	H	-	-	-	-	-	-	-	-	-	-	.	.	.	.	.	.	.	.	.	.	.						
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	.	.	.	.	.	.	.						
16	-	-	-	-	L	-	-	-	L	H	-	-	H	H	H	L	L	.	.	.	.	.	.	.	.	.	.						
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	.	.	.	.	.	.	.						
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	.	.	.	.	.	.	.						
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	.	.	.	.	.	.	.						
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	.	.	.	.	.	.	.						
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	.	.	.	.	.	.	.						
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	.	.	.	.	.	.	.						
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	.	.	.	.	.	.	.						
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	.	.	.	.	.	.	.						
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	.	.	.	.	.	.	.						
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	.	.	.	.	.	.	.						
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	.	.	.	.	.	.	.						
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	.	.	.	.	.	.	.						
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	.	.	.	.	.	.	.						
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	.	.	.	.	.	.	.						
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	.	.	.	.	.	.	.						
FC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	.	.	.	.	.	.	.						
PB	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	A	A	A	A	.	.	.	.	.	.	.						
RB	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	A	A	A	A	.	.	.	.	.	.	.						
LB	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	A	A	A	A	.	.	.	.	.	.	.						
PA	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	A	A	A	A	.	.	.	.	.	.	.						
RA	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	A	A	A	A	.	.	.	.	.	.	.						
LA	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	A	A	A	A	.	.	.	.	.	.	.						
D3	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	A	A	A	A	.	.	.	.	.	.	.						
D2	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	A	A	A	A	.	.	.	.	.	.	.						
D1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	A	A	A	A	.	.	.	.	.	.	.						
D0	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	A	A	A	A	.	.	.	.	.	.	.						
PIN		5	4	3	2	9	8	7	6	19	18	17	16	15	14	13	12	19	18	17	16	15	14	13	12	9	8	7	6				
REMARKS		DAV	TEST	CO	INT/EXT	TBMT	DS	RDV	IDAV	4800 SPS CLK	4800	+32	+16	+8	+4	+2																	

PLS159

T8016605

# Low Cost Programmer for PLD 20-Pin Series

AN12

Table B-2. Low Cost Programmer PLD #U4 Variable Address Generator

CODE NO.		FF MODE										Eb		Ea		POLARITY			REMARKS											
U4		••••••••••										•		•		H H H H														
T E R M	C	AND										(OR)										REMARKS								
		I					B(I)					Q(P)					Q(N)						B(O)							
		3	2	1	0	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	3	2	1	0	
0	-	H	L	L	-	-	H	-	-	H	-	-	-	-	-	-	0	•	•	•	•	•	•	•	•	•	•	INT MODE COUNT		
1	-	H	L	L	-	-	H	-	-	H	-	-	-	-	-	0	•	•	•	•	•	•	•	•	•	•	•	" "		
2	-	H	L	L	-	-	H	-	-	H	-	-	-	-	-	0	-	A	•	•	•	•	•	•	•	•	•	" "		
3	-	H	L	L	-	-	H	-	-	H	-	-	-	-	-	0	-	A	•	•	•	•	•	•	•	•	•	" "		
4	-	H	L	L	-	-	H	-	-	H	-	-	-	-	-	0	-	A	•	•	•	•	•	•	•	•	•	" "		
5	-	H	L	L	-	-	H	-	-	H	-	-	-	-	-	0	-	A	•	•	•	•	•	•	•	•	•	" "		
6	-	H	L	L	-	-	H	-	-	H	-	-	-	-	-	0	-	A	•	•	•	•	•	•	•	•	•	" "		
7	-	H	L	L	-	-	H	-	-	H	-	-	-	-	-	0	-	A	•	•	•	•	•	•	•	•	•	TERM COUNT		
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	A	A	A	A	A			
9	-	H	H	H	L	-	-	H	H	-	-	-	-	-	-	-	-	L	•	•	•	•	•	•	•	•	•	LOAD REBIT J/K		
10	-	H	H	H	L	-	-	H	H	-	-	-	-	-	-	-	-	L	•	•	•	•	•	•	•	•	•	" "		
11	-	H	H	H	L	-	-	H	H	-	-	-	-	-	-	-	-	L	•	•	•	•	•	•	•	•	•	" "		
12	-	H	H	H	L	H	-	-	H	-	-	-	-	-	-	-	-	L	•	•	•	•	•	•	•	•	•	" "		
13	-	H	H	H	H	-	-	H	H	-	-	-	-	-	-	-	-	L	•	•	•	•	•	•	•	•	•	" "		
14	-	H	H	H	H	-	-	H	H	-	-	-	-	-	-	-	-	L	•	•	•	•	•	•	•	•	•	" "		
15	-	H	H	H	H	-	-	H	H	-	-	-	-	-	-	-	-	L	•	•	•	•	•	•	•	•	•	" "		
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	A	A	A	A	A			
17	-	H	L	H	L	L	L	L	H	-	-	-	-	-	-	-	-	H	•	•	•	•	•	•	•	•	•	ONLINE		
18	-	H	L	H	L	L	L	L	H	-	-	-	-	-	-	-	-	L	•	•	•	•	•	•	•	•	•	OFFLINE		
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	A	A	A	A	A			
20	-	H	H	H	L	-	-	L	H	-	-	-	-	-	-	-	-	H	•	•	•	•	•	•	•	•	•	LOAD SET J/K		
21	-	H	H	H	L	-	-	L	H	-	-	-	-	-	-	-	-	H	•	•	•	•	•	•	•	•	•	" "		
22	-	H	H	H	L	-	-	L	H	-	-	-	-	-	-	-	-	H	•	•	•	•	•	•	•	•	•	" "		
23	-	H	H	H	L	-	-	L	H	-	-	-	-	-	-	-	-	H	•	•	•	•	•	•	•	•	•	" "		
24	-	H	H	H	H	-	-	L	H	-	-	-	-	-	-	-	-	H	•	•	•	•	•	•	•	•	•	" "		
25	-	H	H	H	H	-	-	L	H	-	-	-	-	-	-	-	-	H	•	•	•	•	•	•	•	•	•	" "		
26	-	H	H	H	H	-	-	L	H	-	-	-	-	-	-	-	-	H	•	•	•	•	•	•	•	•	•	" "		
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	A	A	A	A	A			
28	-	H	H	L	L	-	-	-	H	-	-	-	-	-	-	-	-	H	H	H	H	H	H	•	•	•	•	TERM CLR. LOWER		
29	-	H	H	L	L	-	-	-	H	-	-	-	-	-	-	-	-	H	H	H	H	H	H	•	•	•	•	TERM CLR. UPPER		
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	A	A	A	A	A	A	A	A			
31	-	H	L	H	L	L	L	L	L	H	-	-	-	-	-	-	-	H	H	H	H	H	H	•	•	•	•	SYNC. RESET		
FC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
PB	•	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
RB	•	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
LB	•	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
PA	•	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
RA	•	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
LA	•	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
D3	-	H	L	L	-	-	-	-	H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
D2	•	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
D1	•	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
D0	-	H	L	L	-	-	-	-	H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			
PIN		5	4	3	2	9	8	7	6	19	18	17	16	15	14	13	12	9	8	7	6									
REMARKS		J DAV	C2	C1	CU	D3-B5C	D2 INC VAR	D1	D0-TERM/LA	ON/OFF LN	VA6	VA5	VA4	VA3	VA2	VA1	VA0													

PLS159

T016705







# Low Cost Programmer for PLD 20-Pin Series

AN12

Table B-5. Low Cost Programmer PLD #U6 Revision "B" Results in New Timing Figures

CODE NO.		FF MODE										E <sub>B</sub>	E <sub>A</sub>	POLARITY			REMARKS										
U <sub>6</sub>		. . . . .										.	.	H	L	L		L									
T E M	C	AND					Q(P)					REMARKS															
		I		B(I)			Q(N)		B(O)			REMARKS															
		3	2	1	0	3	2	1	0	7	6	5	4	3	2	1	0	3	2	1	0	3	2	1	0		
0	-	H	L	-	-	L	H	H	H	H	H	H	L	H	H	H	H									SET INT. MODE	
1	-	L	-	-	-	-	-	-	-	L	H	L	L	H	H	H	H					A	A			SEND INT. MD. C <sub>2</sub> /C <sub>3</sub>	
2	-	L	-	-	-	-	-	-	-	L	H	H	L	H	H	H	H									DELAY	
3	-	H	L	-	H	-	-	-	-	L	H	H	L	H	L	H										SET REP. MD. C <sub>2</sub> /C <sub>3</sub>	
4	-	L	-	-	-	-	-	-	-	L	H	L	L	H	H	H	H									VSP ON	
5	-	L	-	-	-	-	-	-	-	L	H	L	L	H	H	H	H									DELAY	
6	-	L	-	-	-	-	-	-	-	L	H	L	L	L	H	H	L									STROBE ON (DOWN)	
7	-	L	-	-	-	-	-	-	-	L	H	L	L	L	L	H	H									STROBE OFF	
8	-	L	-	-	-	-	-	-	-	L	H	L	L	L	L	H	H									VP OFF	
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	-	L	-	-	-	-	-	-	-	L	H	L	L	H	L	H	H									SET VERIFY MODE	
11	-	L	-	-	-	-	-	-	-	L	H	H	L	L	L	H	H									STROBE ON	
12	-	L	-	-	-	-	-	-	-	L	H	L	L	L	L	H	H									INT. STROBE ON	
13	-	L	-	-	-	-	-	-	-	L	L	H	H	H	L	L	H									INT. STROBE OFF	
14	-	L	-	-	-	-	-	-	-	L	H	H	H	H	L	L	H									STROBE OFF	
15	-	L	L	-	-	-	-	-	-	L	H	H	L	L	L	L	H									INCR. VAR (SEC)	
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	-	L	H	-	-	-	-	-	-	L	H	H	L	H	L	L	H										BAIL OUT
18	A	-	L	-	-	-	-	-	-	L	H	H	L	H	L	L	L										SEND CYCLE COMP
19	-	L	-	-	-	-	-	-	-	L	H	H	L	H	H	L	L										CLEAR INT. MODE
20	-	L	-	-	-	-	-	-	-	H	H	H	L	H	H	L	L										WAIT (EXT. MODE)
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	-	H	L	-	L	-	-	-	-	L	H	H	L	H	H	L	H										SET VFY (ZAP)
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	.	L	-	-	-	-	-	-	-	L	-	-	-	-	-	-	-										SEND INT MD(C <sub>0</sub> )
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	-	H	-	-	-	-	-	-	-	H	H	L	H	H	H	H	H										OFFLINE INITIAL
30	-	H	L	-	-	L	H	L	-	-	-	-	-	-	-	-	-										COMPUTER INITIAL
31	-	-	-	-	-	-	-	-	-	L	L	L	L	L	L	L	L										POWER UP INITIAL
FC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PB	.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RB	.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LB	.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PA	.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RA	.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LA	.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D3	-	-	-	-	-	-	-	-	-	L	L	L	L	L	L	L	L										
D2	-	-	-	-	-	-	-	-	-	L	L	L	L	L	L	L	L										
D1	-	-	-	-	-	-	-	-	-	L	L	L	L	L	L	L	L										
D0	-	-	-	-	-	-	-	-	-	L	L	L	L	L	L	L	L										
PIN		5	4	3	2	9	8	7	6	19	18	17	16	15	14	13	12										
REMARKS		DAY	ONLINE	D3-B5C	D1-ZF	D2 INC. VAR.	C2	C1	CO	INT/EXT	VEY 5 B	MODE DIV	EN	VFSP	UP-DOWN	TD1	TD0										

PLS159

T8017006

## Low Cost Programmer for PLD 20-Pin Series

AN12

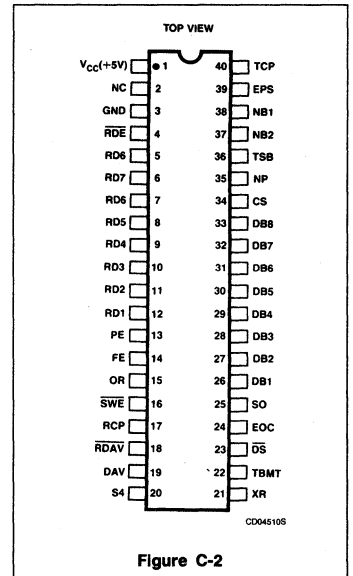
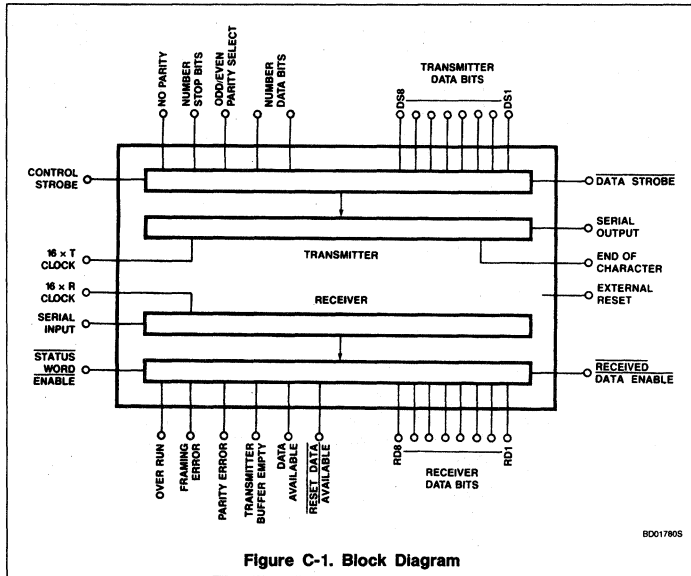
Table B-6. Low Cost Programmer Parts List

REF. DES.	QUANT.	DESCRIPTION	RADIO SHACK #
<b>Resistors</b>			
R1	1	100Ω 1/4W 5%	271-1421
R2	1	620Ω 1/4W 5%	
R3, R5	2	1KΩ 1/4W 5%	271-1324
R14, R15	2	270Ω 1/4W 5%	
R4, R6	2	1.8KΩ 1/4W 5%	271-1307
R7	1	39Ω 1/4W 5%	
R8	1	3.3Ω 1/4W 5%	271-023
R9	1	120Ω 1/4W 5%	
R10	1	75Ω 1/4W 5%	271-023
R11	1	47Ω 1/4W 5%	
R12*	1	1KΩ 1/4W 5%	271-023
R13	1	10KΩ 1/4W 5%	
RN1	1	10KΩ 1/4W 5% 8-Pin SIP	
<b>Capacitors</b>			
C1	1	0.047μF	272-1022
C2, C4	2	4700μF Elec. 35V Axial	
C3	1	220μF Elec. 35V Axial	272-1017
C6, C7	2	470μF	272-1017
C5, C8 - C13	7	0.01μF	
<b>Diodes</b>			
D1A	1	6.2V Zener 1N4735	276-561
D1B	1	9.1V Zener 1N4739	276-563
D2	1	1N4001	276-1101
D3, D4	2	10mA Red LED	276-041
DB1, DB2	2	1.4A Full-Wave Bridge	276-1151
<b>Voltage regulators</b>			
VR1 - VR3	3	LM317 TO-220	276-1778
VR4	1	7805 TO-3	
<b>Integrated circuits</b>			
U1, U2, U4 - U6	5	PLS159IFL — Codes Req'd	276-1794
U3	1	AY-3-1015 UART	
<b>Transformers</b>			
T1, T2	2	18.0VAC CT, 2A	273-1515
FB1	1	Ferrite Bead FB-75B-01 Amidon Associates	
<b>Transistors</b>			
Q1 - Q4	4	2N2222 Transistors	276-2009
Q5	1	TIP 120 Power Dar.	

# Low Cost Programmer for PLD 20-Pin Series

AN12

## APPENDIX C



## AN13 Oscillator With PLS159

### Application Note

#### Application Specific Products

#### INTRODUCTION

The Field-Programmable Logic Sequencer PLS159 may be a self-contained system except that it requires a system clock which is generally provided by an external source. Using a simple R-C network, two bidirectional I/O pins, one can make a simple oscillator which can provide a clock frequency up to 7MHz (typical) with a pulse width of about 75ns. The frequency of oscillation is set by adjusting the RC network, or by adjusting a reference voltage, V. The circuit is shown in Figure 1. This circuit may be used to clock the flip flops on the same chip, or provide a clock for a bigger system. While this circuit uses only two AND-terms and one control term, the rest of the chip may be used for other circuits.

Initially, the capacitor has zero charge and thus is at zero volt. Output is also at a Low, which disables the tri-state buffer. As the capacitor is being charged up, its voltage increases. At about 1.6V, the non-inverting input buffer begins to turn on. One propagation delay later, the output goes High, which enables the tri-state buffer after another propagation delay. Since the tri-state buffer is unconditionally programmed to a Low, it be-

gins to discharge the capacitor at a fast rate. As the input voltage to the non-inverting buffer drops below 1.6V, the buffer begins to change state. One propagation delay later, output goes Low, which disables the tri-state buffer after another propagation delay. The R-C network is ready to start all over again.

To implement this circuit, we first choose  $B_1$  to input voltage from the RC network and  $B_0$  to be the output pin. As shown in Table 1, we enable the output  $B(O)_0$  by "dashing out" the AND-term  $D_0$ , make the output non-inverting by programming POLARITY "H" and connect it to term-0 by entering an A in column  $B(O)_0$  term-0. We then program an H in  $B(I)_1$ , term-0 to connect the non-inverting input of  $B_1$  to term-0. Since we don't need the rest of the inputs and outputs, we "dash out" all other input entries and "dot out" all unused outputs in term-0. To create a Low output on  $B(O)_1$ , we may either have a Low input on the OR and have the output non-inverted, or have a High on the input of OR and have the output inverted to produce a Low. We will arbitrarily use the second method. Here we dash out all inputs and outputs in term-1 except  $B(O)_1$  which is programmed an A. We then program the POLARITY of  $B(O)_1$  "L" to be inverting.

Finally we program an H in  $B(I)_0$  and term- $D_1$  and dash out all unused inputs so that when  $B(O)_0$  becomes High,  $D_1$  will enable the output buffer  $B(O)_1$ . The program is further illustrated by Figure 2.

Since the High output pulsewidth is about two propagation delays of  $B_1$  to  $B_0$ , it is wide enough to drive the clock input of the PLS159. If a square wave is desired, one of the flip flops may be programmed to toggle, as shown in Table 1, term-4, in which case the output frequency of the square wave is half that of  $B(O)_0$ .

The stability of the output frequency can be controlled by:

1. Making V a stable voltage source,
2. making the charging current of the RC network much greater than the input current of  $B(I)_1$  (see Figure 4 for input current characteristics),
3. making the period of oscillation much greater than four times the internal propagation delay (typical  $t_{PD} = 35ns$ ).

#### Recommendations:

- a) Use capacitance less than 500pf.
- b) Choose R such that  $V/R \gg I_{IL}$  (since  $I_{IL} = -100\mu A$ , worst case, over temperature range,  $V/R$  may be 1mA, for example).

The recommendations above put a limit to the range of frequency which the PLS159 may operate in exchange for stable operating frequency. One may extend its range by reducing C to zero (remove the capacitor) and thus operate the circuit at its maximum frequency, or increase the resistance to infinite (remove the resistor), and let the external capacitor be charged by the input current  $I_{IL}$ , which gives a much lower frequency.

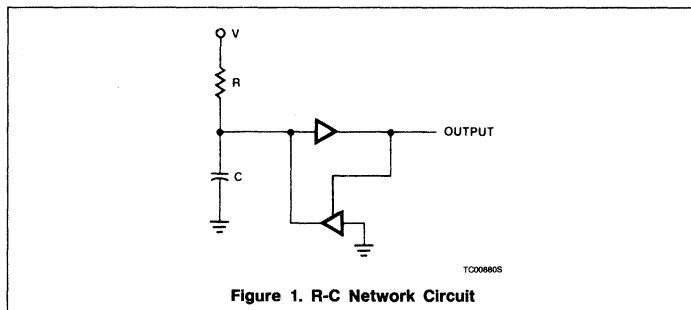


Figure 1. R-C Network Circuit

# Oscillator With PLS159

AN13

Table 1. PLS159 FPLS Program Table

CODE NO.		FF MODE										E <sub>B</sub>										E <sub>A</sub>										POLARITY		
																																L H		
TERM	C	AND										REMARKS	(OR)										REMARKS											
		B(I)					Q(P)						Q(N)					B(O)																
		3	2	1	0	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	3	2	1	0					
0	-	-	-	-	-	-	H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*	*	*	A	B <sub>0</sub> = B <sub>1</sub>					
1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	*	*	A	*	B <sub>0</sub> = LOW					
2																																		
3																																		
4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	*	*	*	SQUARE WAVE					
5																													SEE COMMENT #2					
6																																		
7																																		
8																																		
9																																		
10																																		
11																																		
12																																		
13																																		
14																																		
15																																		
16																																		
17																																		
18																																		
19																																		
20																																		
21																																		
22																																		
23																																		
24																																		
25																																		
26																																		
27																																		
28																																		
29																																		
30																																		
31																																		
FC																																		
PB																																		
RB																																		
LB																																		
PA																																		
RA																																		
LA																																		
D3																																		
D2																																		
D1	-	-	-	-	-	-	H	-	-	-	-	-	-	-	-																			
D0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-																			
PIN		5	4	3	2	9	8	7	6	19	18	17	16	15	14	13	12	19	18	17	16	15	14	13	12	9	8	7	6					
REMARKS																																		

PLS159

TB00510S

**COMMENTS:**

- D<sub>1</sub> is controlled by B(1)<sub>0</sub>, which enables B(O)<sub>1</sub> when B(O)<sub>0</sub> outputs a High.
- Q(N)<sub>0</sub> is programmed as J-K flip-flop, and will toggle automatically at each rising-edge of the clock.

**NOTE:**

- In order for the flip-flop to work the clock out (B<sub>0</sub>) must be externally connected to pin 1.

# Oscillator With PLS159

AN13

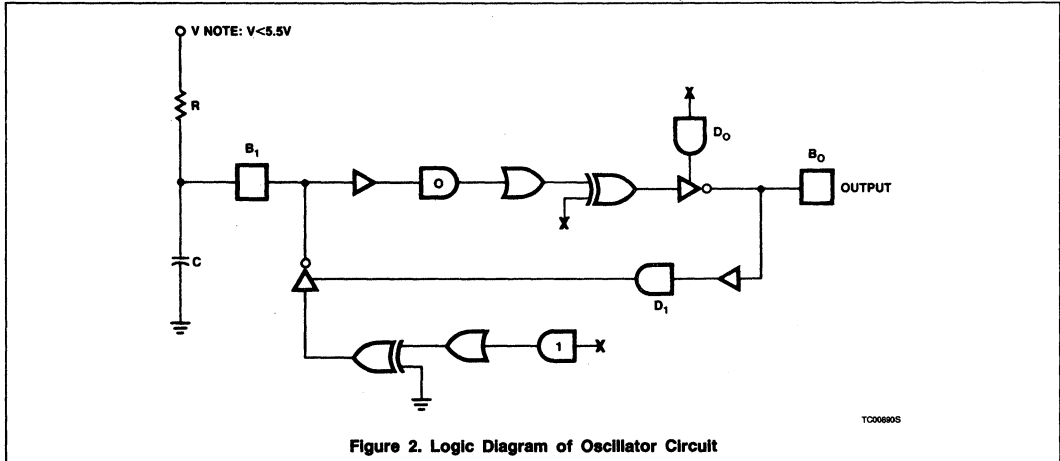
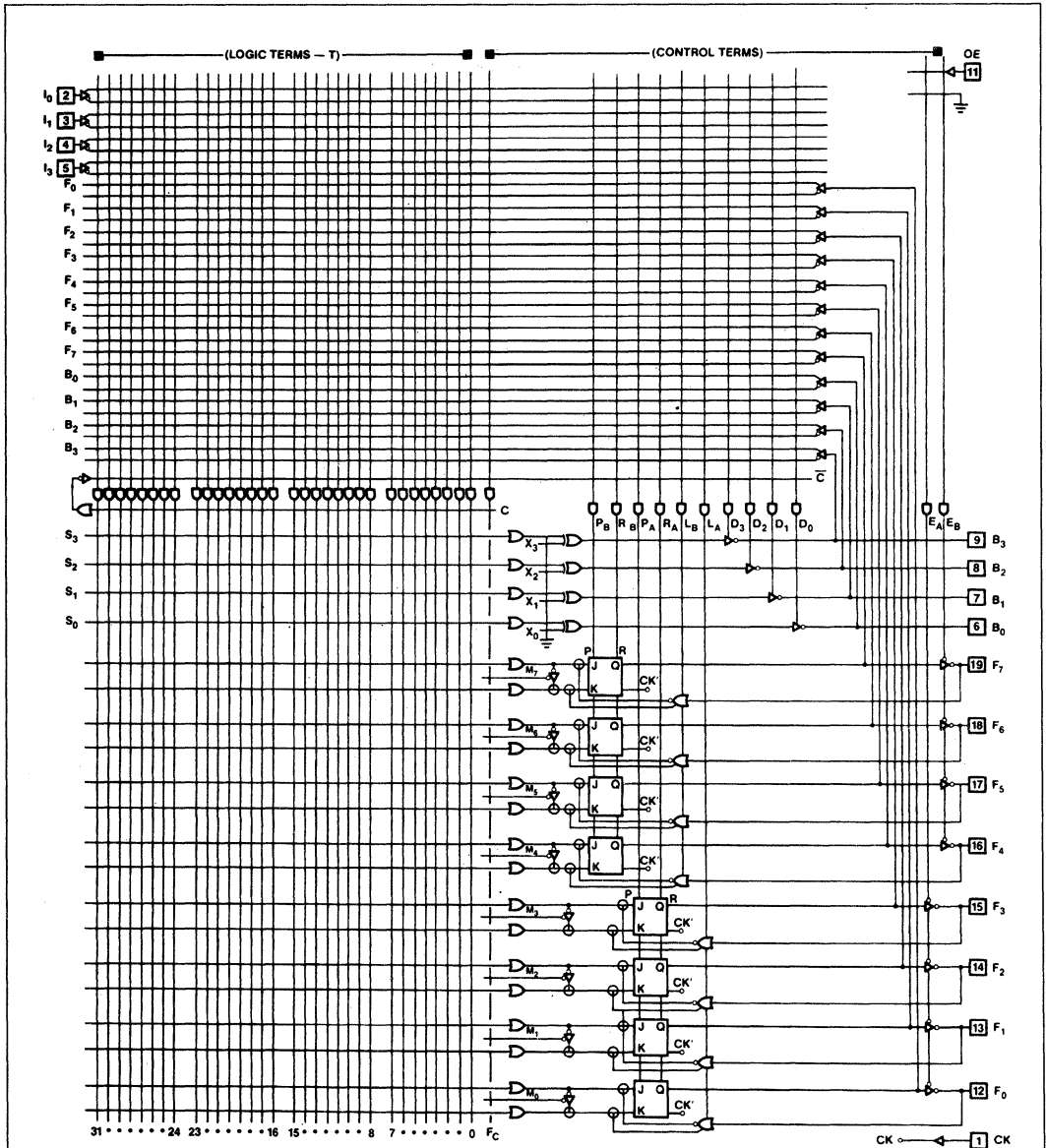


Figure 2. Logic Diagram of Oscillator Circuit

# Oscillator With PLS159

# AN13



- NOTES:**
1. All OR gate inputs with a blown link float to logic "0".
  2. All other gates and control inputs with a blown link float to logic "1".
  3. @ denotes WIRE-OR.

Figure 3. Logic Diagram of PLS159 FPLS

LD016618

# Oscillator With PLS159

AN13

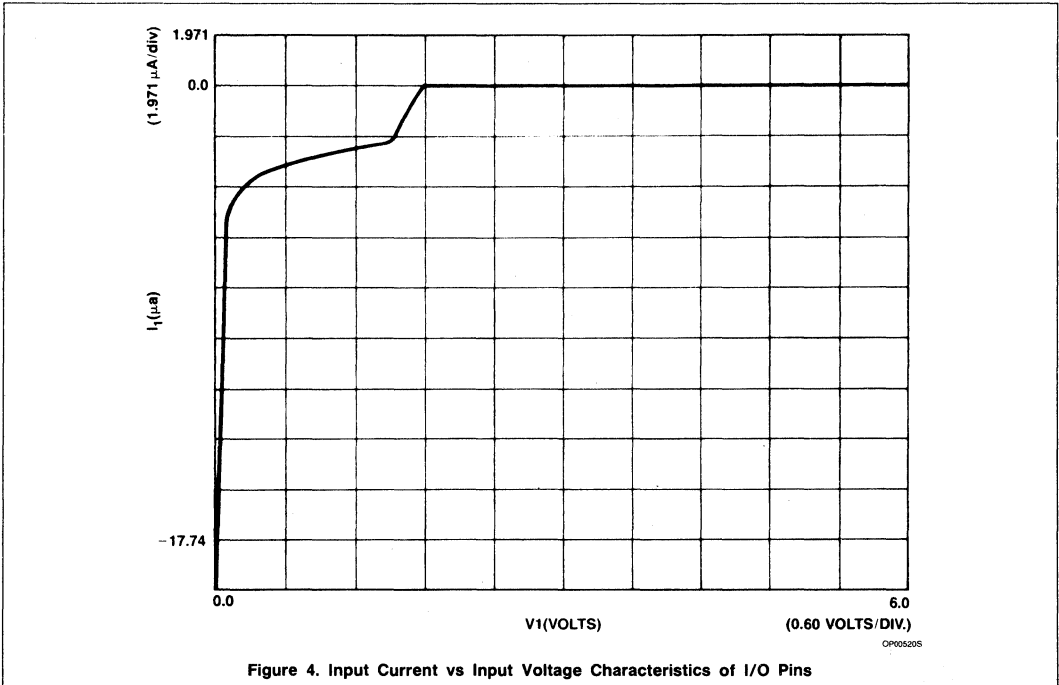


Figure 4. Input Current vs Input Voltage Characteristics of I/O Pins

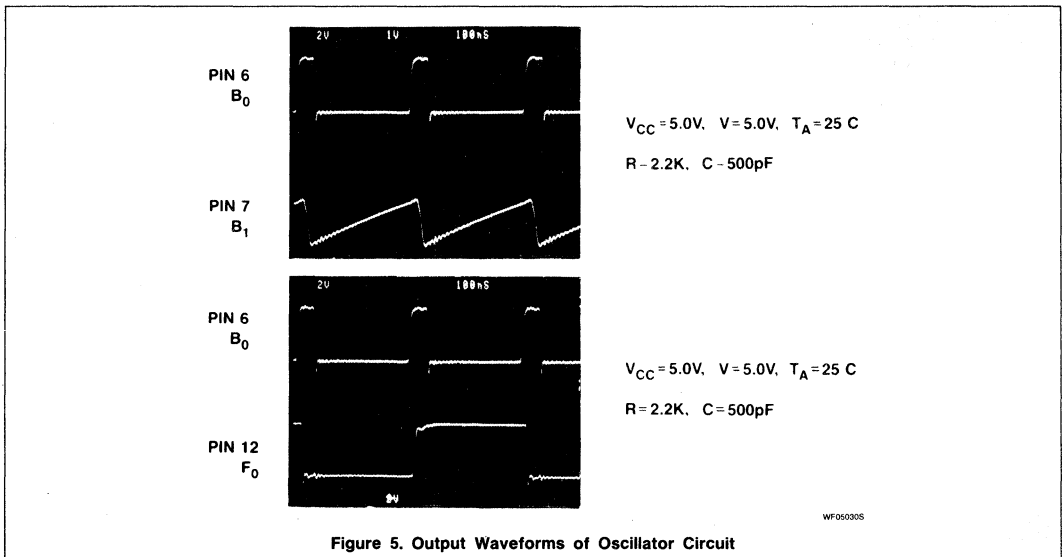


Figure 5. Output Waveforms of Oscillator Circuit





## AN14 Latches and Flip-Flops With PLS153

### Application Note

#### Application Specific Products

#### DESCRIPTION

Using the simple AND, OR and INVERT logic functions of the PLS153, memory functions such as latches and edge-triggered flip-flops may be implemented with a relatively small part of the chip and without external wiring. In this application note, we will discuss the implementation of two R-S latches, a D-latch, an edge-triggered R-S flip-flop, and an edge-triggered D flip-flop.

#### INTRODUCTION TO PLS153

To implement this function, let's first take a look at the PLS153 logic diagram and its programming table as shown in Appendices A and B. On the left side of the logic diagram (Appendix A) are eight dedicated inputs,  $I_0$  to  $I_7$ , each of which has a true and a complement output. Each output is connected to the inputs of 32 AND functions (we will call them AND-terms from now on), the outputs of which are, in turn, connected to the inputs of ten OR functions. The output of each OR function is connected to one input of an Exclusive-OR function, which is in turn connected to a non-inverting output buffer. The function of the XOR is to control the output polarity. The output, in its virgin state, is non-inverting, since one side of the XOR is

connected to ground by the fuse  $X_n$ , where  $n = 0, 1, 3...9$ . To have the output inverted, one needs only to blow fuse  $X_n$  open so that the  $X_n$  input is unconditionally High. The output buffers are all three-state buffers which may be enabled or disabled by their corresponding AND gates. The output buffers are disabled in their virgin state. All pins labeled "B" are bidirectional. Input buffers of the "B" pins are identical to those of the "I" pins.

The programming table shown in Appendix B emulates a truth table. All the inputs to the device are positioned on the left side, and all the outputs are on the right side. Each row in the table corresponds to an 18-input AND-term with up to ten outputs. On the left side, or the input side of the table, each column represents an input. The 18 columns represent input buffers  $I_0$  to  $I_7$ ,  $B_0$  to  $B_9$ . To distinguish between inputs and outputs of the bidirectional pin, B(I) is used for input and B(O) is used for outputs as shown in the programming table. On the right side of the table, each column represents an output circuit (B(O) $_{0-9}$ ) which consists of an OR gate, an XOR, and a non-inverting three-state buffer. The output buffers are controlled by AND-terms  $D_0$  to  $D_9$ , the inputs of which may be connected to any number of the 18 inputs.

The polarity of the outputs is defined by the POLARITY entries which are on the upper right corner of the programming table.

To program the inputs to the AND-terms, an "H" will cause the fuse of the inverting input buffer to be blown, leaving the non-inverting buffer connected to the AND-term; an "L" will do the opposite. A "-" will cause both fuses to be blown, and therefore the programmed input is a "Don't care". A "0", the virgin state of the device, has both fuses intact, which causes the output of the AND-term to be unconditionally Low.

To program the outputs, a "." causes the fuse that connects the output of AND-term to the input of an OR to be blown and thus renders the output inactive. An "A" causes the fuse to remain intact and thus the output is active.

The output polarity of each output buffer may be programmed by entering an "H" or an "L" in the POLARITY section. An "L" causes the XOR to blow its grounding fuse and become inverted, whereas an "H" leaves the fuse intact and the output is non-inverted.

To AND several inputs, we put them in a row; to OR several inputs, we put them in different rows, as shown in illustrations in Appendix B.

# Latches and Flip-Flops With PLS153

AN14

## SIMPLE R-S LATCH

A simple R-S latch may be formed by cross-coupling two NAND functions together as shown in Figure 1.

As an illustration, let's assign the input R to I<sub>0</sub> of the PLS153, input S to I<sub>1</sub>, output Q to B<sub>0</sub>, and output  $\bar{Q}$  to B<sub>1</sub>. As shown in Table 1, to form the NAND gates we need to program the POLARITY Low on B(O)<sub>0</sub> and B(O)<sub>1</sub>. To unconditionally enable the output buffers, we "dash" out all inputs to D<sub>0</sub> and D<sub>1</sub>. As for the inputs, we put an "H" on I<sub>0</sub>, term-0 for the input R, non-inverted; another "H" on B(I)<sub>1</sub>, term-0 for the feedback from Q. In the same manner, we program I<sub>1</sub>, term-1 and B(O)<sub>0</sub> "H". The POLARITY, rows 0, 1, D<sub>0</sub> and D<sub>1</sub>, forms a "truth table" with which one can analyze his own or someone else's design. The program in Table 1 may be illustrated as shown in Figure 2.

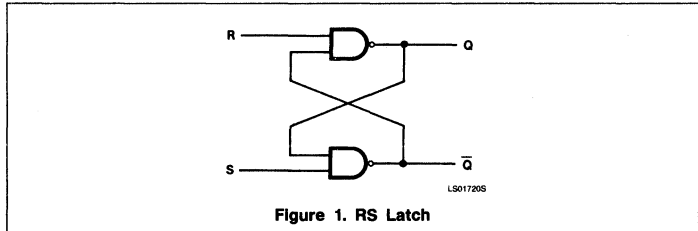


Figure 1. RS Latch

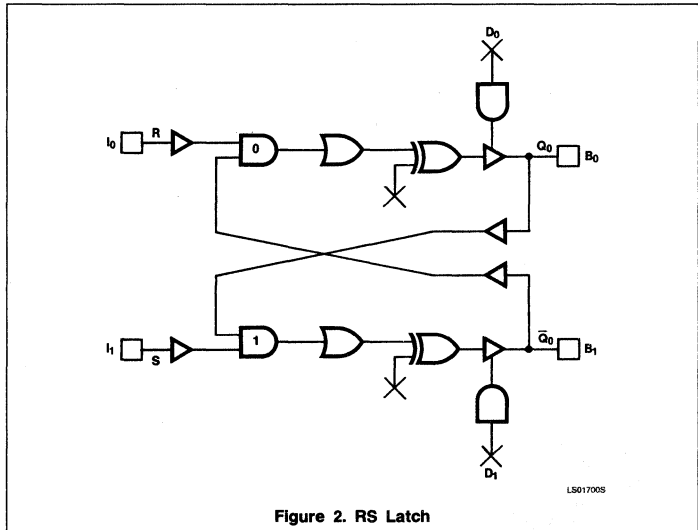


Figure 2. RS Latch



# Latches and Flip-Flops With PLS153

AN14

## ANOTHER SIMPLE R-S LATCH

Another way to implement a simple latch is shown in Figure 3, in which two NOR functions are cross-coupled to form a latch.

As with the previous example, we first define the input and output pins. For this example, we use  $I_2$  for the R input,  $I_3$  for the S input,  $B_2$  for the Q output, and  $B_3$  for the  $\bar{Q}$  output. We program  $B_2$  and  $B_3$  to have inverted outputs by programming POLARITY of  $B_2$  and  $B_3$  Low, as shown in Table 2. Terms 6 and 7 are ORed together by  $B(O)_2$ , rows 6 and 7. In the same manner  $B(O)_3$  ORs Terms 8 and 9. The programmed table of this design may be represented as shown in Figure 4.

Since each AND-term of the PLS153 can accommodate up to 18 inputs (true or inverting inputs of eight from  $I_0$  to  $I_7$  and ten from  $B_0$  to  $B_9$ ), and each OR circuit can be connected to up to thirty-two AND-terms, we can add additional features such as those shown in Figure 5.

The programming of this design is left to the reader as an exercise.

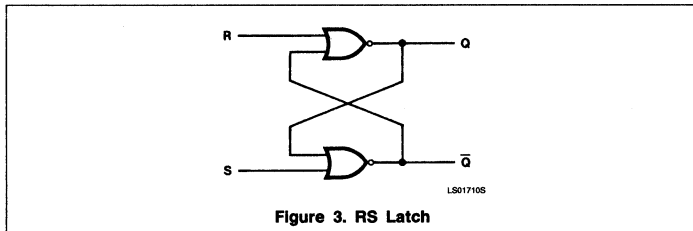


Figure 3. RS Latch

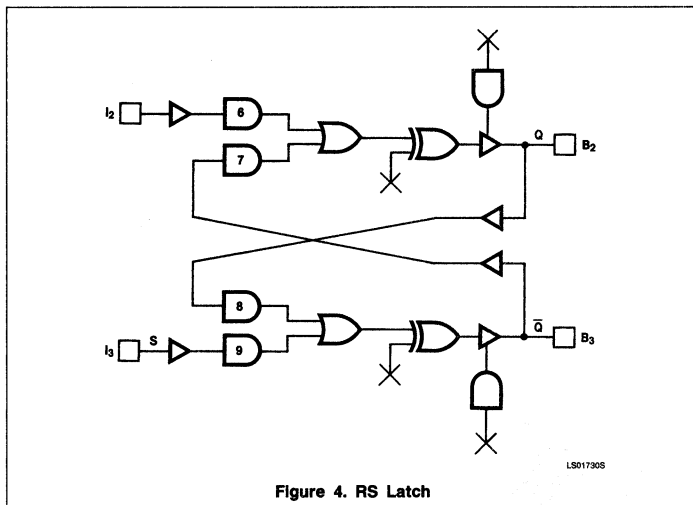


Figure 4. RS Latch

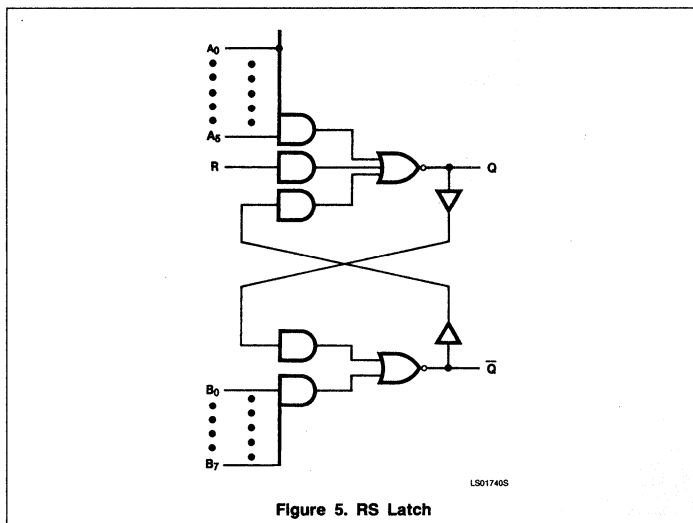


Figure 5. RS Latch



# Latches and Flip-Flops With PLS153

AN14

## D-LATCH

A simple D-latch can be constructed with an PLS153 as shown in Figure 6.

This circuit may be easily programmed into the PLS153 as shown in Table 3. The program may, in turn, be represented as shown in Figure 7.

This circuit may be expanded to have multiple D-latches using the same latch enable (LE).

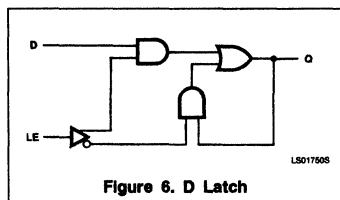


Figure 6. D Latch

## R-S FLIP-FLOP

Two R-S latches may be combined to form a master-slave flip-flop that is triggered at the rising-edge of the clock (or the falling-edge of the clock, if the designer so desires). Figure 8 shows a combination of two sets of cross-coupled NOR gates concatenated to form the flip-flop. The implementation of this circuit using PLS153 is as illustrated in Table 4 and Figure 9.

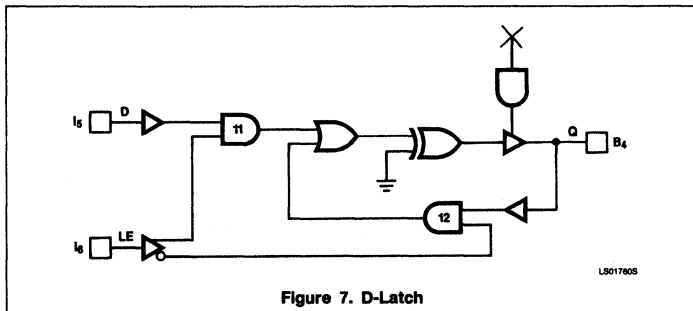


Figure 7. D-Latch

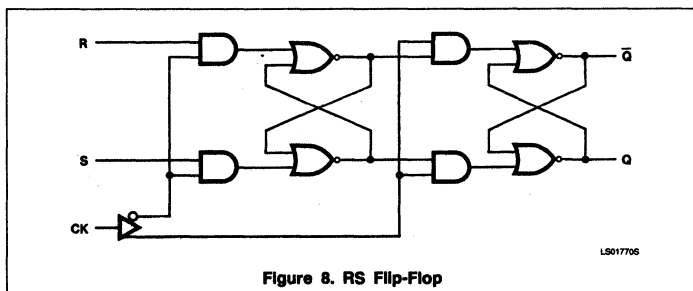


Figure 8. RS Flip-Flop

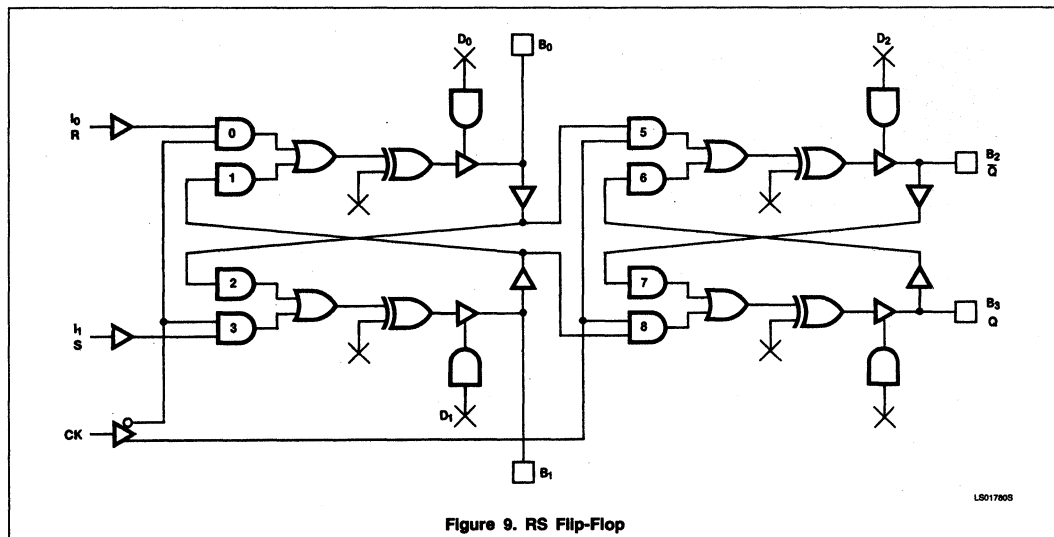


Figure 9. RS Flip-Flop



# Latches and Flip-Flops With PLS153

AN14

Table 4. PLS153/153A Programming Table

CODE NO.													POLARITY																
T	AND										REMARKS	OR					REMARKS												
	B(I)											B(O)																	
M	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	
0	-	-	-	-	L	H	-	-	-	-	-	-	-	-	-	-	-	$\bar{C}K \cdot R$	•	•	•	•	•	•	•	•	A	)	$(R \cdot \bar{C}K + B_0)$
1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	H	-	-	$B_1$	•	•	•	•	•	•	•	•	A	)	
2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	H	-	$B_0$	•	•	•	•	•	•	•	•	A	)	$(S \cdot \bar{C}K + B_0)$
3	-	-	-	-	L	H	-	-	-	-	-	-	-	-	-	-	-	$\bar{C}K \cdot S$	•	•	•	•	•	•	•	•	A	)	
4																													
5	-	-	-	-	H	-	-	-	-	-	-	-	-	-	-	H	-	$\bar{C}K \cdot B_0$	•	•	•	•	•	•	•	•	A	)	$(B_0 \cdot \bar{C}K + B_0)$
6	-	-	-	-	-	-	-	-	-	-	-	-	-	H	-	-	-	$B_3$	•	•	•	•	•	•	•	•	A	)	
7	-	-	-	-	-	-	-	-	-	-	-	-	-	H	-	-	-	$B_2$	•	•	•	•	•	•	•	•	A	)	$(B_1 \cdot \bar{C}K + B_2)$
8	-	-	-	-	H	-	-	-	-	-	-	-	-	H	-	-	-	$\bar{C}K \cdot B_1$	•	•	•	•	•	•	•	•	A	)	
9																													
10																													
11																													
12																													
13																													
14																													
15																													
16																													
17																													
18																													
19																													
20																													
21																													
22																													
23																													
24																													
25																													
26																													
27																													
28																													
29																													
30																													
31																													
D9																													
D8																													
D7																													
D6																													
D5																													
D4																													
D3																													
D2																													
D1																													
D0																													
PN	8	7	6	5	4	3	2	1	19	18	17	16	15	14	13	12	11	9	19	18	17	16	15	14	13	12	11	9	
REMARKS																													

PLS153



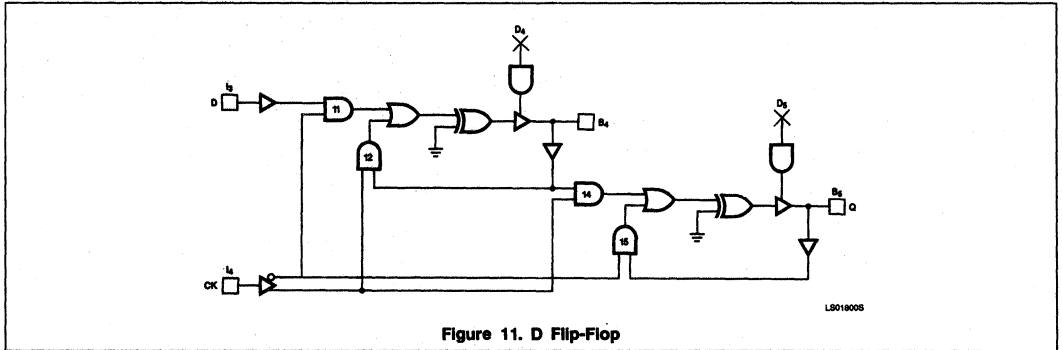
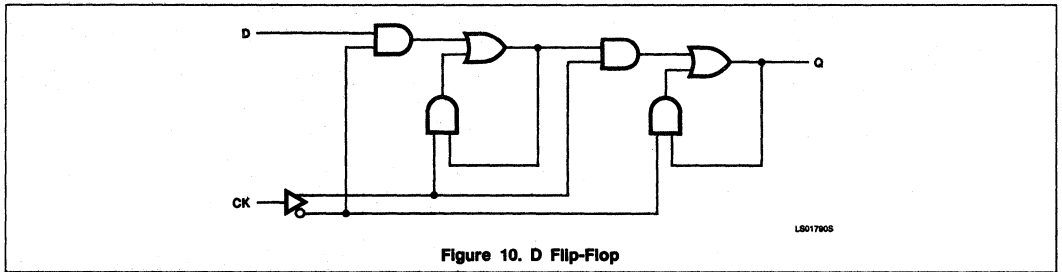
# Latches and Flip-Flops With PLS153

AN14

## D FLIP-FLOP

An edge-triggered master-slave D flip-flop may be constructed with two D-latches in the manner shown in Figure 10.

An PLS153 may be programmed as shown in Figure 11 to implement the D flip-flop which is equivalent to the circuit shown in Table 5 in the PLS153 logic representation.



# Latches and Flip-Flops With PLS153

AN14

Table 5. PLS153/153A Programming Table

CODE NO.													POLARITY																															
AND													OR																															
B(I)													B(O)																															
T	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	REMARKS	REMARKS														
0	-	-	-	-	L	-	H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	R · CK	· · · · · · · · · · A	(R · CK + B <sub>4</sub> )						
1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	B <sub>1</sub>	· · · · · · · · · · A							
2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	B <sub>0</sub>		A						
3	-	-	-	-	L	H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	S · CK		A						
4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
5	-	-	-	-	-	H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CK · B <sub>0</sub>		A	(B <sub>0</sub> · CK + D)					
6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	Q		A						
7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	B <sub>2</sub>		A	(B <sub>1</sub> · CK + B <sub>2</sub> )					
8	-	-	-	-	-	H	-	-	-	-	-	-	-	-	-	-	H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	B <sub>1</sub> · CK		A						
9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
11	-	-	-	L	H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	$\overline{CK} \cdot D$		A	( $\overline{CK} \cdot D + CK \cdot$					
12	-	-	-	H	-	-	-	-	-	-	-	-	-	-	-	H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CK · B <sub>4</sub>		A	B <sub>4</sub> )					
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
14	-	-	-	H	-	-	-	-	-	-	-	-	-	-	-	H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CK · B <sub>4</sub>		A	(CK · B <sub>4</sub> + $\overline{CK} \cdot$					
15	-	-	-	L	-	-	-	-	-	-	-	-	-	-	-	H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	$\overline{CK} \cdot Q$		A	D)					
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
22	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
23	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
24	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
25	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
26	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
27	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
28	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
29	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
30	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
31	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
D9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
D8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
D7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
D6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
D5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
D4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
D3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
D2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
D1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
D0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-									
PN	8	7	6	5	4	3	2	1	19	18	17	16	15	14	13	12	11	9	19	18	17	16	15	14	13	12	11	9																
REMARKS				CK	D	CK	S	R					B <sub>5</sub> (Q)	B <sub>4</sub>	Q = B <sub>3</sub>	Q = B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>																										

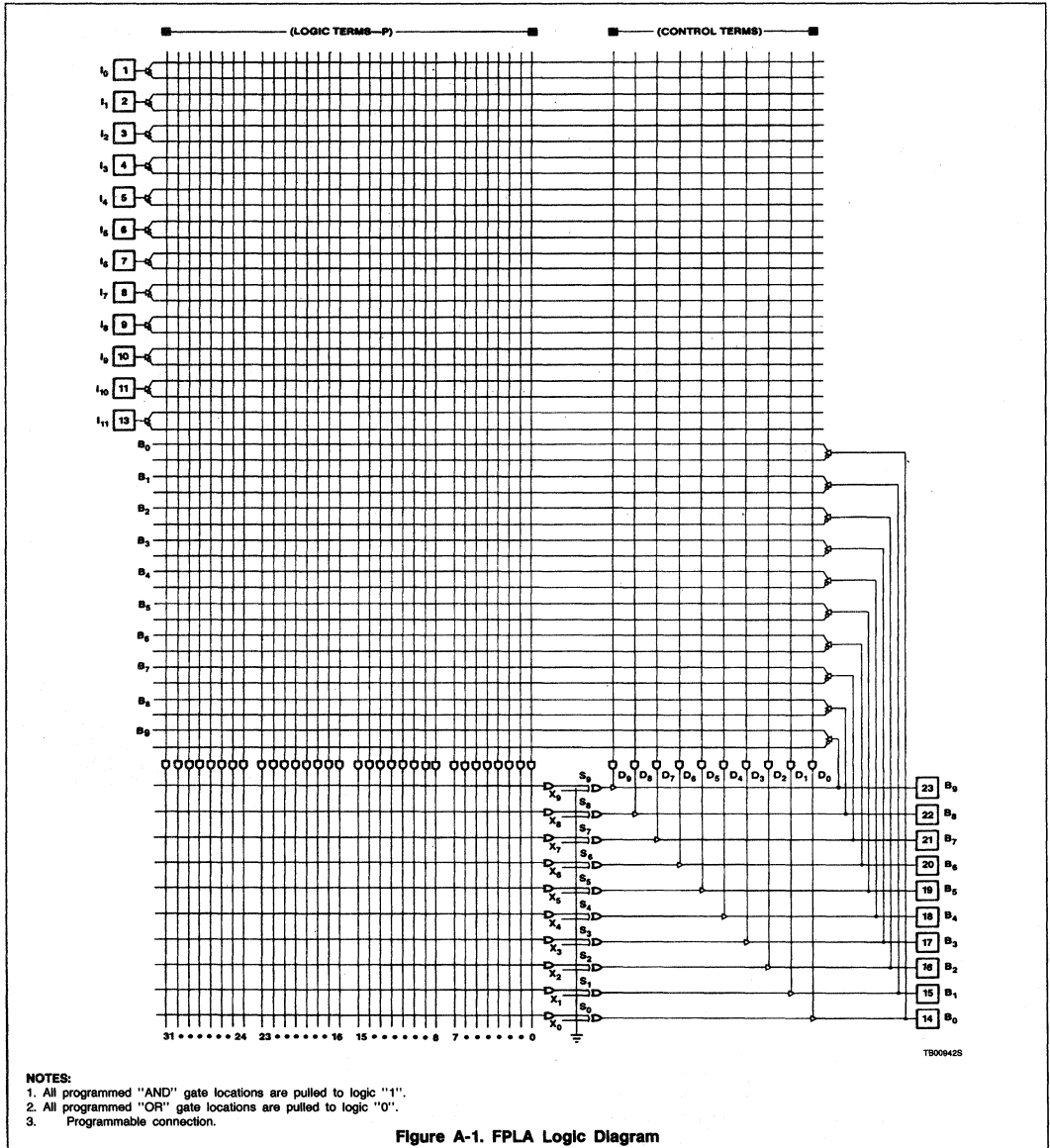
PLS153

T1009506

# Latches and Flip-Flops With PLS153

AN14

## APPENDIX A



# Latches and Flip-Flops With PLS153

AN14

Appendix B. PLS153/153A Programming Table

CODE NO.													POLARITY								REMARKS								
T M	AND												OR																
	B(I)												B(O)																
	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0											
0	H	H	H	H	-	-	-	-	-	-	-	-	-	-	-	-	-	A	•	•	•	A	•	•	•	•	•	X = A • B • C • D	
1																												T = / (A • B • C • D)	
2																													
3																													
4	L	L	L	L	-	-	-	-	-	-	-	-	-	-	-	-	-	A	•	•	•	A	•	•	•	•	•	Y = A • B • C • D	
5																												U = / (A • B • C • D)	
6																													
7																													
8																													
9	-	H	L	-	-	-	-	-	-	-	-	-	-	-	-	-	-	B	•	C	•	A	•	•	A	•	•	Z = (B • C + B • D)	
10	-	H	L	-	-	-	-	-	-	-	-	-	-	-	-	-	-	B	•	D	•	A	•	•	A	•	•	T • D	
11	-	-	L	L	-	-	-	-	-	-	-	-	-	-	-	-	-	C	•	D	•	A	•	•	A	•	•	V = / (B • C + B • D + C • D)	
12																													
13																													
14																													
15																													
16																													
17																													
18																													
19																													
20																													
21																													
22																													
23																													
24																													
25																													
26																													
27																													
28																													
29																													
30																													
31																													
D9																													
D8																													
D7																													
D6																													
D5																													
D4																													
D3																													
D2																													
D1																													
D0																													
PN	8	7	6	5	4	3	2	1	19	18	17	16	15	14	13	12	11	9	19	18	17	16	15	14	13	12	11	9	
REMARKS	A	B	C	D				X	Y	Z			T	U	V				X	Y	Z			T	U	V			

PLS153

# Latches and Flip-Flops With PLS153

AN14

## APPENDIX C

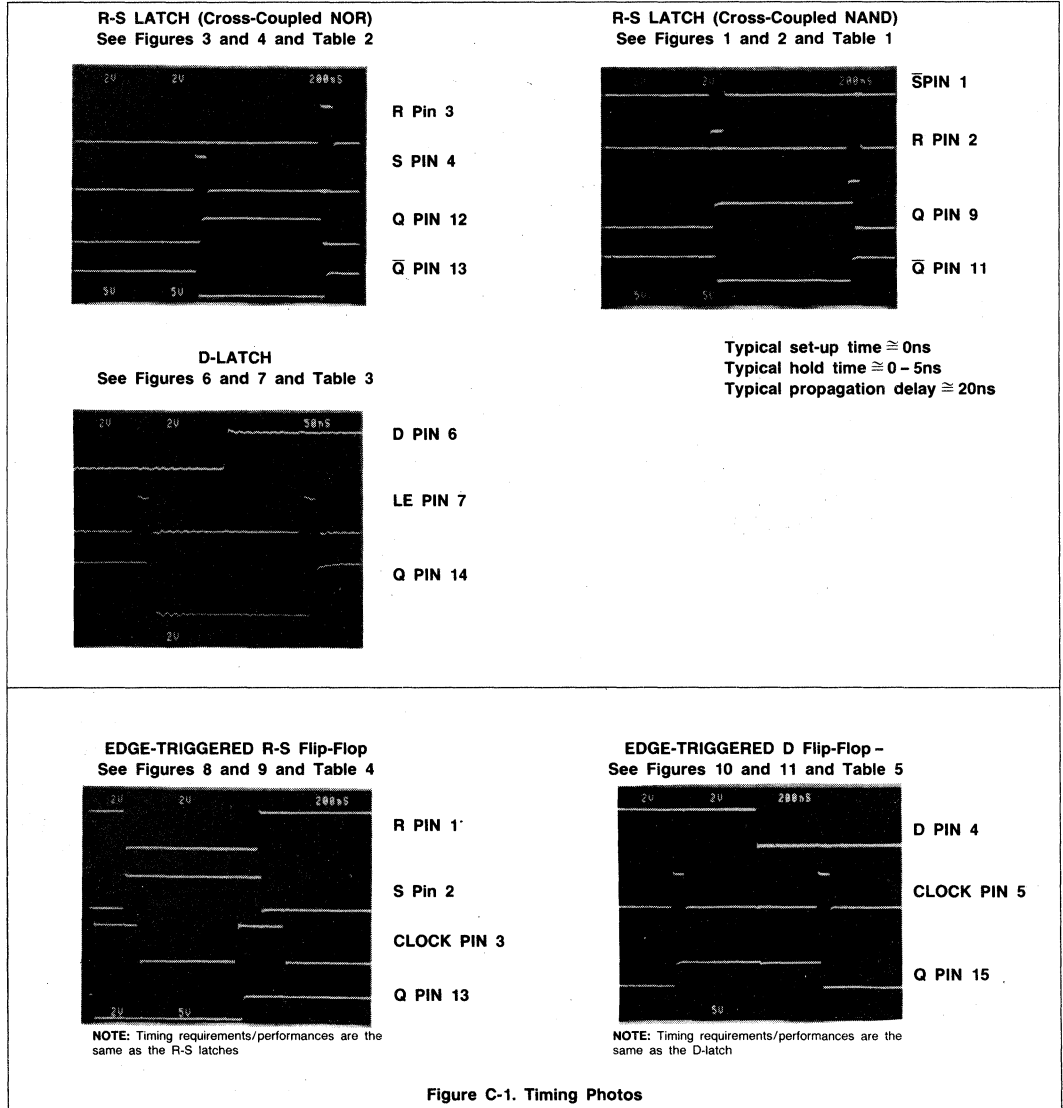


Figure C-1. Timing Photos

#### Application Specific Products

#### INTRODUCTION

The PLS159 is a field programmable logic sequencer which consists of four dedicated inputs, four bidirectional I/O's, eight flip-flops, thirty two 16-input AND gates, twenty 32-input OR gates, and a complement array. Each flip-flop has a bidirectional I/O and may be individually programmed as J-K or D flip-flop, or switch between the two types dynamically. The flip-flops will accept data from the internal logic array or from the bidirectional I/O, or they may be set or reset asynchronously from the AND array. The output polarity of the four bidirectional I/O's are programmable and the direction is controlled by the AND array. Figure 1 is the logic diagram of PLS159.

#### PROGRAMMING THE PLS159

The programming table is shown in Table 1 where there is a place for everything that is

shown in Figure 1. The program table is basically divided into two main sections. The left hand side of the table, section A, represents the input side of the AND gates, while the right hand side, section B, represents the OR gates sections which includes the flip-flops and the combinatorial outputs B(0) to B(3). The flip-flops modes are defined in section C and the output polarities of the combinatorial outputs are defined in section E. The programming symbols are detailed in Figure 2.

As shown in Table 1, the programming table is very similar to a truth table. Each column in section A represents an input to the 32 AND gates, and each row represents an AND gate connecting to 17 inputs. Columns I<sub>0</sub> to I<sub>3</sub> represent the 4 dedicated inputs, I<sub>0</sub> to I<sub>3</sub>. Columns B(I)<sub>0</sub> to B(I)<sub>3</sub> represent the inputs of the 4 bidirectional I/O, B<sub>0</sub> to B<sub>3</sub>. Columns

Q(P)<sub>0</sub> to Q(P)<sub>7</sub> represent the feedback, F<sub>0</sub> to F<sub>7</sub>, from the flip-flops (the present state). Column "C" represents the complement array.

As shown in Figure 1, the outputs of the AND gates are connected to an array of OR gates which, in turn, are connected to either flip-flops or output circuits. Columns Q(N)<sub>0</sub> to Q(N)<sub>7</sub> represent the next state which the flip-flops will be in. Columns B(O)<sub>0</sub> to B(O)<sub>3</sub> represent the combinatorial outputs B<sub>0</sub> to B<sub>3</sub>.

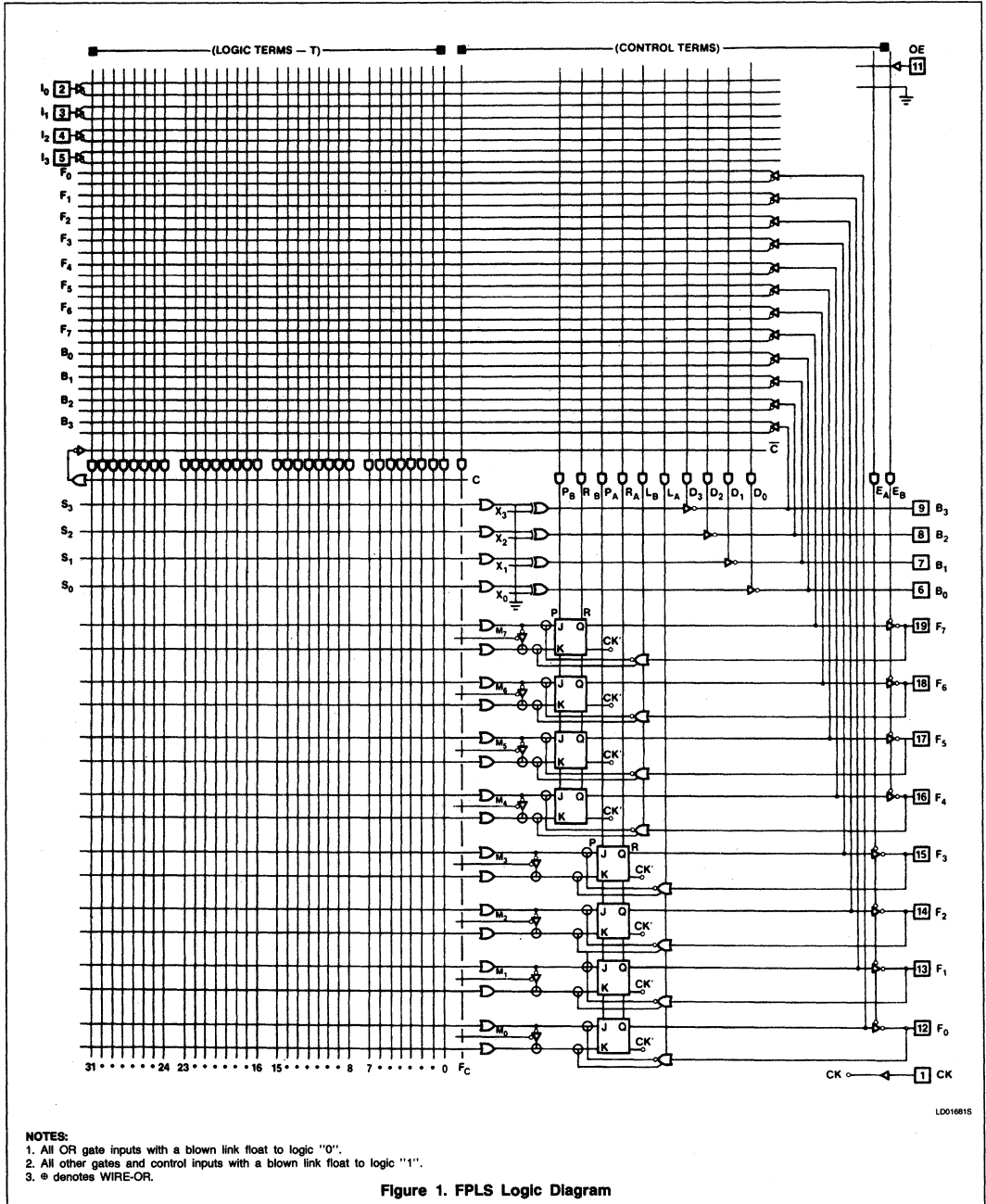
Each row represents an AND gate with 17 inputs each of which may be true and/or complement and is, therefore, a perfect decoder. Referring to the programming symbols in Figure 2, to implement the equation

$$Z = A * B * C * D,$$

all one has to do is to enter one line as shown in Table 2, term-0.

PLS159 Primer

AN15



- NOTES:
1. All OR gate inputs with a blown link float to logic '0'.
  2. All other gates and control inputs with a blown link float to logic '1'.
  3.  $\odot$  denotes WIRE-OR.

Figure 1. FPLS Logic Diagram

LD016815

PLS159 Primer

AN15

Table 1. FPLS Program Table

CODE NO.		FF MODE														EB		EA		POLARITY									
SECTION C																SECTION D						SECTION E							
TERM	C	AND										REMARKS	(OR)						REMARKS										
		I			B(I)			Q(P)					Q(N)			B(O)													
		3	2	1	0	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	3	2	1	0
0																													
1																													
2																													
3																													
4																													
5																													
6																													
7																													
8																													
9																													
10																													
11																													
12																													
13																													
14																													
15																													
16																													
17																													
18																													
19																													
20																													
21																													
22																													
23																													
24																													
25																													
26																													
27																													
28																													
29																													
30																													
31																													
FC																													
PB																													
RB																													
LB																													
PA																													
RA																													
LA																													
D3																													
D2																													
D1																													
D0																													
PIN		5	4	3	2	9	8	7	6	19	18	17	16	15	14	13	12	19	18	17	16	15	14	13	12	9	8	7	6
REMARKS																													

PLS159

TB000518



PLS159 Primer

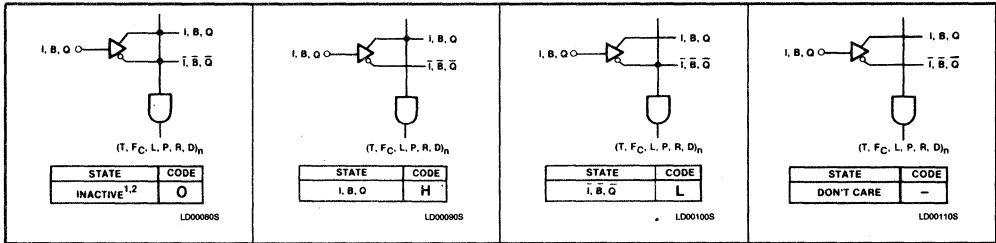
AN15

The FPLS can be programmed by means of Logic Programming equipment.

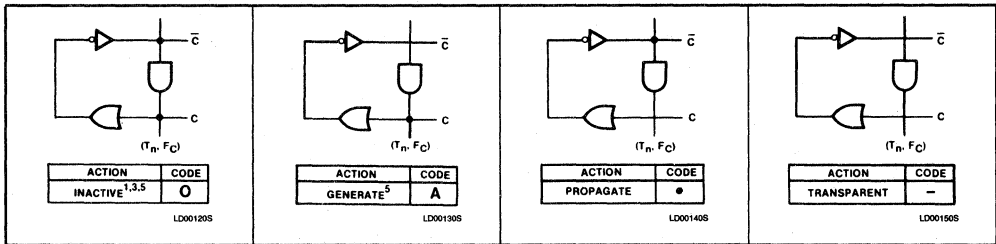
With Logic programming, the AND/OR-EX-OR input connections necessary to imple-

ment the desired logic function are coded directly from the State Diagram using the Program Tables on the following pages.

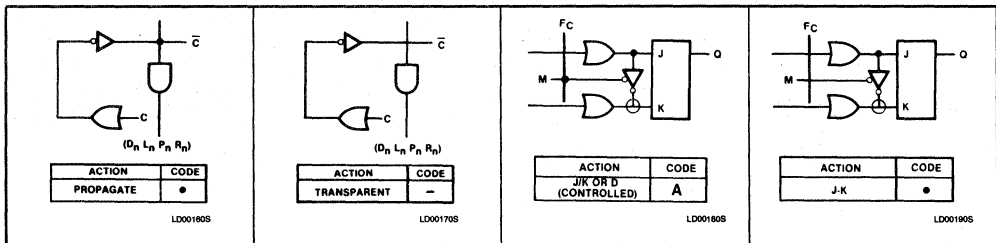
In these Tables, the logic state or action of all I/O, control, and state variables is assigned a symbol which results in the proper fusing pattern of corresponding links defined as follows:



"AND" ARRAY - (I), (B), (Qp)

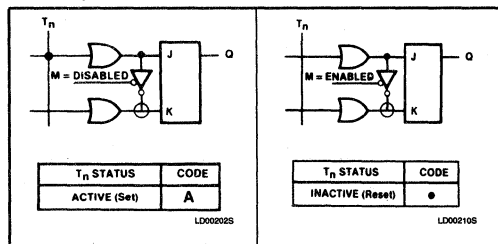


"COMPLEMENT" ARRAY - (C)



"COMPLEMENT" ARRAY (cont.)

"OR" ARRAY - (MODE)

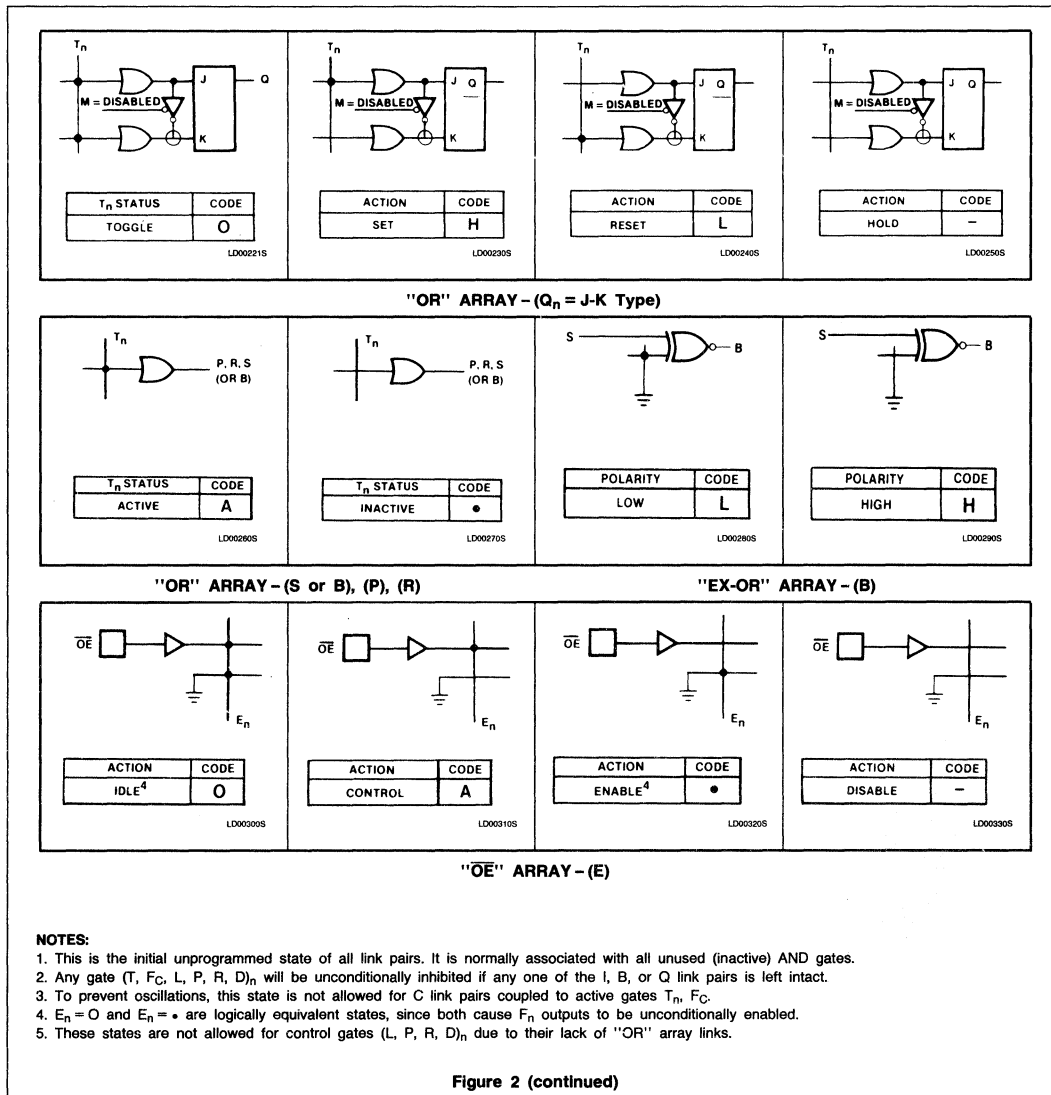


"OR" ARRAY - (Q<sub>N</sub> = D-Type)

Figure 2

PLS159 Primer

AN15





## PLS159 Primer

## AN15

Notice that only  $I_0$  to  $I_3$  on the left hand side and  $B(O)_4$  on the right hand side have entries to implement the equation. All unused columns are dashed out or dotted out.

To implement the equation

$$Y = /A * B * /C,$$

enter one line as shown in Table 2, term-2 where the entry "H" represents the non-inverting input buffer while the entry "L" represents the inverting buffer. To have the AND gate to be unconditionally "High", dash out all the inputs of that particular AND gate as shown in Table 2, term-4. The virgin condition of the device, as shipped from the factory, has all connections intact, which means that the inverting and the non-inverting buffers of the same inputs are connected together. Such connection will cause the AND gate to be unconditionally "Low" as shown in Table 2, terms 6 and 7. The unconditional High and Low states are normally useful only internally and seldom brought out to the output pins.

To implement the equation

$$W = A * /B + C * /D,$$

enter one line for  $A * /B$  and another line for  $C * /D$  as shown in Table 2, terms 9 and 10. Use one line to AND something together; use different lines to OR something together — one line per item to be OR'ed.

All the pins which are labelled B's are bidirectional I/O pins. Their input buffers are represented by the B(I) columns on the left hand side of the programming table. An "H" entry represents the non-inverting buffer and an "L" entry represents the inverting buffer. Their output buffers are represented by the B(O) columns on the right hand side of the table. An "A" entry means that the output is active (connected to the AND gates); a "."

entry means that the output is inactive (not connected). The outputs may be programmed to be inverting or noninverting. The polarity of each output is determined by its exclusive OR gate (Figure 1 and Figure 2). To have a non-inverting output, enter an "H" in the section labelled "POLARITY" (Table 1, Section E). To have an inverting output, enter an "L". For example, Table 3, terms-0 and-2 implement the equation

$$Z = /(A * B) \text{ and } Y = A * B$$

respectively. The above two equations may also be implemented by term-4 which uses the same AND gate to drive two OR gates.

Besides being able to have programmable Active-High or Active-Low output, the programmable output polarity feature also allows the user to minimize his AND term utilization by converting his logic equation into other forms such as conversion by De Morgan Theorem.

For example, the equation

$$X = A + B + C + D$$

takes four AND terms to implement as shown in Table 3, terms 6 to 9. By using De Morgan Theorem, the same equation is changed to

$$/W = /A * /B * /C * /D$$

The result is as shown in term 11 — a saving of three AND terms. The output buffers are disabled in their virgin states so that they all behave as inputs. The buffers are enabled or disabled by their corresponding Control AND terms  $D_0$  to  $D_3$  (see Figure 1). The Control AND terms are represented in the programming table on the last four rows on the left hand side. Dashing out all the inputs will

cause the output buffer to be unconditionally enabled, whereas a "0" (zero) will cause the buffer to be unconditionally disabled. The buffers may also be controlled by a logical condition, e.g.  $A * /B * /C$ , etc.

There are eight flip-flops on the chip each of which may be programmed as a J/K or a D flip-flop, or they may be programmed to switch dynamically. As shown in Figure 1, each flip-flop is a J/K to begin with. A tri-state inverter is connected in between the J and K inputs of each flip-flop, which when enabled by the AND gate  $F_C$ , will cause the flip-flop to function as a D flip-flop. The inverters are enabled by  $F_C$  through fuses  $M_0$  to  $M_7$ . A "." in the F/F Mode entry of the programming table means that particular fuse is to be disconnected and that particular flip-flop is to be J/K. An "A" entry will leave the M fuses intact, which allow the flip-flop to be D or J/K as controlled by the output of  $F_C$  (see Figure 2, "OR" ARRAY — (MODE)). The inputs to the flip-flops are represented by the programming table as the next state,  $Q(N)_{0 \text{ to } 7}$  since their inputs are from the OR array. The outputs of these registers are connected to their respective tri-state inverting output buffers, four of which are controlled by EA and the other four by EB. A "." in EA will enable outputs  $F_0$  to  $F_3$ , whereas a "." will disable them. An "A" will allow the output buffers to be controlled by /OE, pin 11. Table 4, terms 0, 1 and 3 represent the following equations

$$Q_0: J = A * C + /B * /E \quad \text{eq. 1}$$

$$Q_0: K = A * /C \quad \text{eq. 2}$$

Notice that the J input in equation 1 is represented by the "H" entry in terms-0 and 1, column  $Q(N)_0$  while the K input in equation 2 is represented by the "L" entry in term-3, column  $Q(N)_0$ . An undefined input, J or K, is considered "Low".





# PLS159 Primer

# AN15

A D flip-flop may be implemented by first entering an "A" in F/F MODE. Then enter "0" in the row F<sub>C</sub>, which will unconditionally enable the tri-state inverter between the J and K inputs. The following logic equation may be implemented as shown in Table 4, term 5

$$Q_1: D = /A * /B * /C + E.$$

Notice that the entries in term 5, columns Q(N)<sub>0</sub> to 7 are "A" and "." instead of "H" and "L" as in the case of J/K flip-flops. The entry "A" will cause the fuse connecting to the "K" input to be disconnected and the "J" fuse to be intact. Whereas the entry "." will cause both fuses to be disconnected. This feature enables the user to quickly recognize the mode in which the flip-flops are operating without having to go through the control terms. Some commercially available device programmers in the market may not have the software capability to implement this feature, in which case an "H" and a "." may be used in place of "A" and "." respectively as shown in Table 4, terms 8 and 9.

Of course, the term F<sub>C</sub> may have inputs instead of zeros and dashes, in which case the flip-flop modes are controlled dynamically.

When both the J and K inputs are "1's", the flip-flop will toggle. A simple 3-bit counter may be implemented using only AND terms as shown in Table 4 terms 11, 12 and 13. The logic equations for the three flip-flops are as the following:

$$\begin{aligned} Q_5: T &= 1; && (Q_5 \text{ toggles unconditionally}) \\ Q_6: T &= Q_5; && (Q_6 \text{ toggles when } Q_5 = 1) \\ Q_7: T &= Q_5 * Q_6; && (Q_7 \text{ toggles when } Q_5 * \\ &&& Q_6 = 1) \end{aligned}$$

The above equations represent an octal up-counter. However, since the outputs of the flip-flops are inverted, the counting sequence of the outputs is that of a down-counter.

The flip-flops may be asynchronously set and reset by the Control AND terms PA/PB and

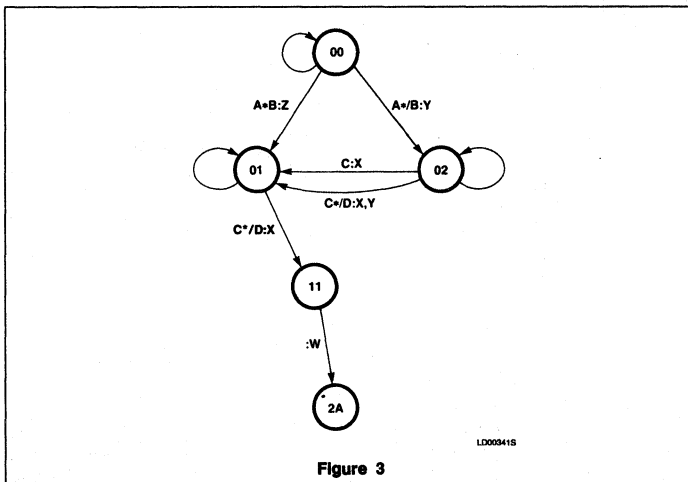


Figure 3

L0000416

RA/RB respectively. As shown in Figure 1, PA and RA controls flip-flops F<sub>0</sub> to F<sub>3</sub>, while PB and RB control F<sub>4</sub> to F<sub>7</sub>.

In order to save the number of input pins, the eight flip-flops may be synchronously loaded directly from their own output pins. To use this feature, EA and/or EB must be programmed "A" or "." so that the output buffers may be disabled before loading. As shown in Figure 1, every flip-flop has an OR/NOR gate the input of which is directly connected to the output pin and the outputs of the OR/NOR are connected to the K and J inputs respectively. This OR/NOR gate inverts the input and feeds it to the flip-flop in a "wire-OR" fashion. Therefore, when loading data directly into the flip-flops from the output pins, caution must be exercised to insure that the inputs from the OR array does not interfere with the data being loaded. For example, if the data being loaded is a "1" on the output

pin, the J input will be a "0" and the K input will be a "1". If, at the same time, a "1" is present at the J-input from the OR array, the flip-flop will see "1's" in both J and K inputs. It will toggle as a result. The OR/NOR gates are enabled by the Control AND terms LA and LB. LA controls flip-flops F<sub>0</sub> to F<sub>3</sub> and LB controls F<sub>4</sub> to F<sub>7</sub>.

All Control AND terms function and are programmed in the same manner as the other AND terms. The only difference is that the Control AND terms are not connected to the OR array.

The outputs of the flip-flops may be fed back into the AND array as the present state, Q(P). The output of the AND array into the OR array and the inputs to the flip-flops is the next state, Q(N). As an example, Figure 3 is a state machine implemented in a PLS159 as shown in Table 5, terms 0 to 6.





# AN16 PLS159 As Modulo Counters

## Application Note

### Application Specific Products

#### DESCRIPTION

The field-programmable logic sequencer (FPLS) PLS159 has eight edge-triggered flip-flops which may be individually programmed to be J/K, D, or toggle flip-flops. The detail logic diagram of the device is shown in Figure 2.

A counter may be constructed by having individual flip-flops toggle when they are at certain states. Implementing such counters requires one toggle flip-flop per bit and an associating AND gate as a decoder for each flip-flop. Such a method allows the PLS159 to be a counter from modulo 2 to 256 or from one bit to 8 bits.

#### MODULO-8 UP/DOWN COUNTER

##### The Up-Counter

The PLS159 has an inverting output buffer for each flip-flop. Therefore the counting sequence of the flip-flops is from 7 to 0 so that the outputs will count from 0 to 7. Table 1 is

the truth table of the internal counting sequence.

Table 1

COUNT	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>
0	1	1	1
1	1	1	0
2	1	0	1
3	1	0	0
4	0	1	1
5	0	1	0
6	0	0	1
7	0	0	0

From Table 1, logic equations for the flip-flops may be generated as the following:

$$\begin{aligned}
 Q_0: T &= 1; && (Q_0 \text{ unconditionally toggles}) \\
 Q_1: T &= /Q_0; && (Q_1 \text{ toggles if } Q_0 = 0) \\
 Q_2: T &= /Q_0 * /Q_1; && (Q_2 \text{ toggles if both } Q_0 \text{ and } Q_1 = 0)
 \end{aligned}$$

The equations are implemented as shown in Table 2, terms 0, 1, and 2. E<sub>A</sub> is enabled, and the mode for the three flip-flops is set to J/K. All unused flip-flops and outputs (the B(0)'s) associating to AND-terms are "dashed" or "dotted" out. The event to be counted is connected to the clock input, pin 1 of the device.

##### The Down-Counter

The down-counter is constructed in the same manner as the up-counter except that the counting sequence is reversed. The down-counter is implemented in Table 2, terms 4 to 6 with the corresponding flip-flop mode set to J/K and E<sub>B</sub> enabled.

##### Modulo-16 Counter

Expanding the modulo-8 counter to 4-bit, the counter becomes modulo-16, Table 3, terms 0 to 3 and the associating F/F MODE and E<sub>A</sub> implement a modulo-16 up-counter, while terms 5 to 8 and the associating F/F MODE and E<sub>B</sub> implement the modulo-16 down-counter.



# PLS159 As Modulo Counters

AN16

Table 3

CODE NO.		FF MODE										E <sub>B</sub>		E <sub>A</sub>		POLARITY		REMARKS										
T E R M	C	AND										(OR)										REMARKS						
		I			B(I)			Q(P)				Q(N)					B(O)											
		3	2	1	0	3	2	1	0	7	6	5	4	3	2	1	0		7	6	5		4	3	2	1	0	
		0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	.		.	.	.	.	.	.
1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	.	.	.	.	.	.	.			
2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	.	.	.	.	.	.	.			
3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	.	.	.	.	.	.	.			
4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	.	.	.	.	.	.	.			
5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	-	.	.	.	.	.	.	.			
6	-	-	-	-	-	-	-	-	-	-	-	-	-	H	-	-	0	-	.	.	.	.	.	.	.			
7	-	-	-	-	-	-	-	-	-	-	-	-	H	H	-	-	0	-	.	.	.	.	.	.	.			
8	-	-	-	-	-	-	-	-	-	H	H	H	-	-	-	-	0	-	.	.	.	.	.	.	.			
9																												
10																												
11																												
12																												
13																												
14																												
15																												
16																												
17																												
18																												
19																												
20																												
21																												
22																												
23																												
24																												
25																												
26																												
27																												
28																												
29																												
30																												
31																												
FC																												
PB																												
RB																												
LB																												
PA																												
RA																												
LA																												
D3																												
D2																												
D1																												
D0																												
PIN	5	4	3	2	9	8	7	6	19	18	17	16	15	14	13	12	19	18	17	16	15	14	13	12	9	8	7	6
REMARKS																												

PLS159

# PLS159 As Modulo Counters

AN16

## DECADE COUNTER

### The Decade Up-Counter

The truth table of a decade counter is as shown in Table 4.

Table 4

COUNT	Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>
0	1	1	1	1
1	1	1	1	0
2	1	1	0	1
3	1	1	0	0
4	1	0	1	1
5	1	0	1	0
6	1	0	0	1
7	1	0	0	0
8	0	1	1	1
9	0	1	1	0
0	1	1	1	1

Logic equations may be generated from Table 4 as the following:

$$\begin{aligned}
 Q_0: T &= 1; \\
 Q_1: T &= /Q_0 * Q_3; \\
 Q_2: T &= /Q_0 * /Q_1 * Q_3; \\
 Q_3: T &= /Q_0 * /Q_1 * /Q_2 * Q_3 + \\
 &\quad /Q_0 * Q_1 * Q_2 * /Q_3;
 \end{aligned}$$

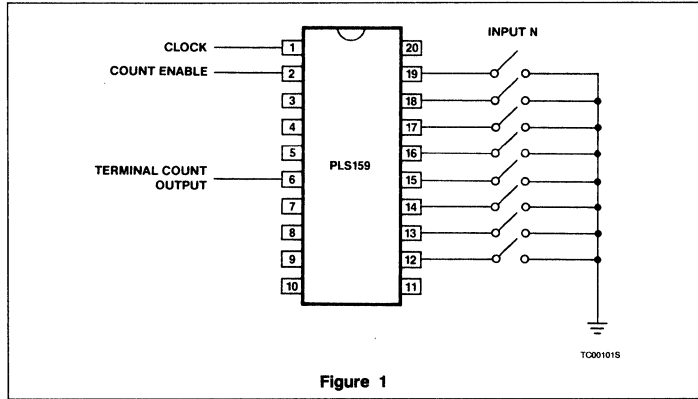


Figure 1

The equations are implemented as shown in Table 5, terms 0 to 4 and the associating flip-flop mode and E<sub>A</sub> settings.

### Decade Down-Counter

The decade down-counter is similar to the decade up-counter except that the counting sequence is reversed from 0 to 9 as shown in Table 5, terms 6 to 11 with the associating F/F MODE set to J/K and E<sub>B</sub> to ENABLE.

### Decade Down Counter

COUNT	Q <sub>7</sub>	Q <sub>6</sub>	Q <sub>5</sub>	Q <sub>4</sub>
9	0	1	1	0
8	0	1	1	1
7	1	0	0	0
6	1	0	0	1
5	1	0	1	0
4	1	0	1	1
3	1	1	0	0
2	1	1	0	1
1	1	1	1	0
0	1	1	1	1
9	0	1	1	0

$$\begin{aligned}
 Q_4: T &= 1; \\
 Q_5: T &= Q_4 * Q_5 * Q_6 * /Q_7 + \\
 &\quad Q_4 * /Q_5 * /Q_6 * Q_7 + \\
 &\quad Q_4 * Q_5 * /Q_6 * Q_7 + \\
 &\quad Q_4 * /Q_5 * Q_6 * Q_7; \\
 Q_6: T &= Q_4 * Q_5 * Q_6 * /Q_7 + \\
 &\quad Q_4 * Q_5 * /Q_6 * Q_7; \\
 Q_7: T &= Q_4 * Q_5 * Q_6 * /Q_7 + \\
 &\quad Q_4 * Q_5 * Q_6 * Q_7;
 \end{aligned}$$

### DIVIDE-BY-N COUNTER

This counter, shown in Figure 1 and Table 6, takes an input (maximum = 8 bits for each device), complements it, and counts until all flip-flops become 1, then a terminal count is sent out at B(O)<sub>0</sub> (divide-by-N). At the end of the count, where all flip-flops are 1's, terms LA and LB are activated which causes the flip-flops to load data from F<sub>0</sub> to F<sub>7</sub>, and the cycle starts again. Since there are 8 flip-flops, the maximum count is 256.

# PLS159 As Modulo Counters

# AN16

Table 5

CODE NO.			FF MODE								E <sub>B</sub>		E <sub>A</sub>		POLARITY															
			••••••••								•		•																	
T E R M	AND										REMARKS	(OR)										REMARKS								
	C	I			B(I)			Q(P)				Q(N)					B(O)													
	3	2	1	0	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	3	2	1	0		
0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	•	•	•	•	•	•	•	•	•	•	•	•		
1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	•	•	•	•	•	•	•	•	•	•	•	•		
2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	•	•	•	•	•	•	•	•	•	•	•	•		
3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	•	•	•	•	•	•	•	•	•	•	•	•		
4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	•	•	•	•	•	•	•	•	•	•	•	•		
5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	•	•	•	•	•	•	•	•	•	•	•	•		
6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	•	•	•	•	•	•	•	•	•	•	•	•		
7	-	-	-	-	-	-	-	-	L	H	H	H	-	-	-	0	0	0	0	-	-	-	-	-	-	•	•	•		
8	-	-	-	-	-	-	-	-	H	L	L	H	-	-	-	0	0	0	-	-	-	-	-	-	•	•	•			
9	-	-	-	-	-	-	-	-	H	L	H	H	-	-	-	0	0	0	-	-	-	-	-	-	•	•	•			
10	-	-	-	-	-	-	-	-	H	H	L	H	-	-	-	0	0	0	-	-	-	-	-	-	•	•	•			
11	-	-	-	-	-	-	-	-	H	H	H	H	-	-	-	0	0	0	-	-	-	-	-	-	•	•	•			
12																														
13																														
14																														
15																														
16																														
17																														
18																														
19																														
20																														
21																														
22																														
23																														
24																														
25																														
26																														
27																														
28																														
29																														
30																														
31																														
FC																														
PB																														
RB																														
LB																														
PA																														
RA																														
LA																														
D3																														
D2																														
D1																														
D0																														
PIN	5	4	3	2	9	8	7	6	19	18	17	16	15	14	13	12	19	18	17	16	15	14	13	12	9	8	7	6		
REMARKS																														

PLS159

TB000325

# PLS159 As Modulo Counters

# AN16

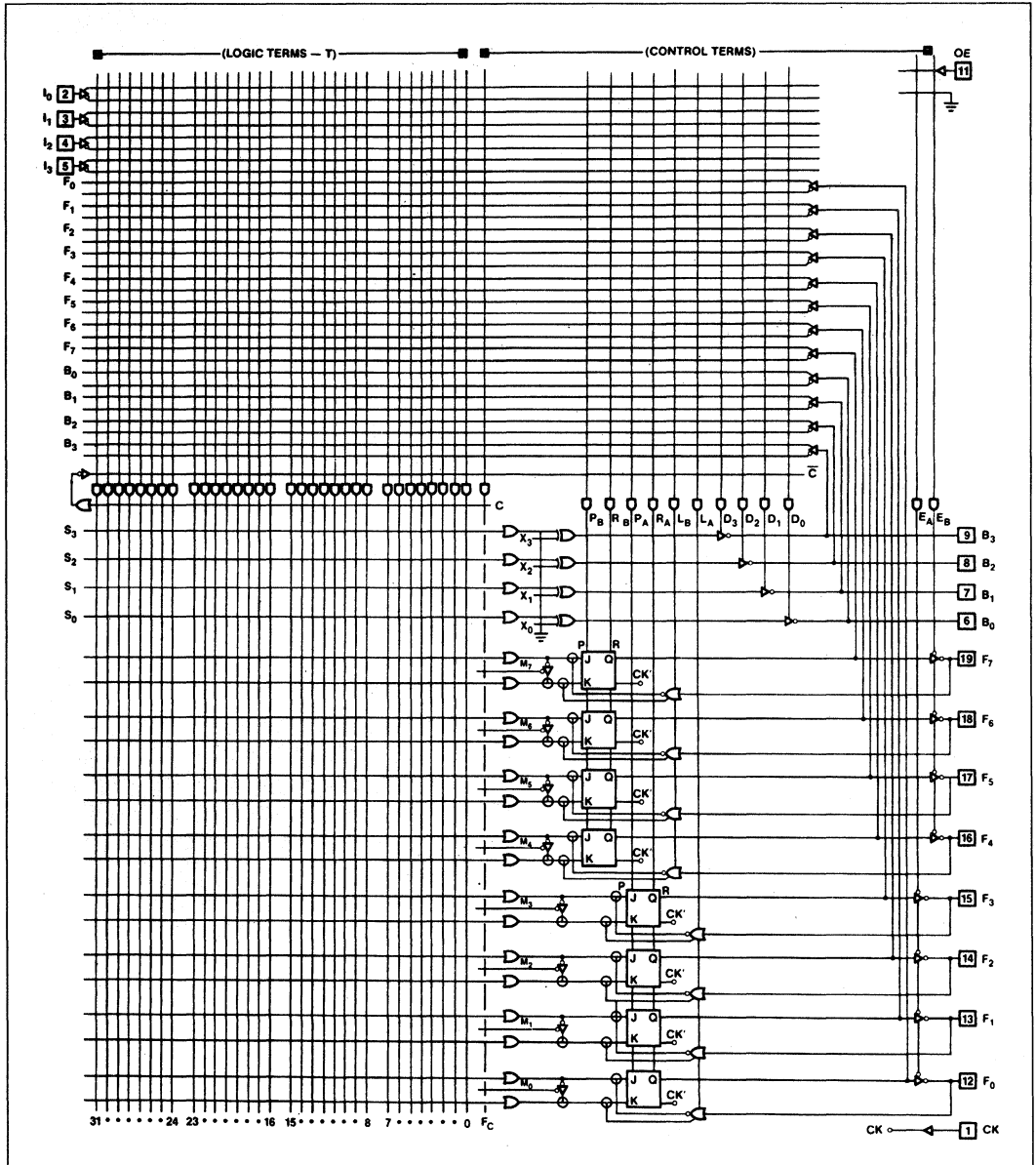
Table 6

TERM	CODE NO.											FF MODE											REMARKS	E <sub>B</sub>				E <sub>A</sub>				POLARITY		REMARKS
												•••••••••••												-				-				H		
	AND											Q(N)												(OR)										
C	I			B(I)			Q(P)						Q(N)				B(O)																	
	3	2	1	0	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	3	2	1	0						
0	-	-	-	L	-	-	-	L	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	.	.	.	.	.					
1	-	-	-	L	-	-	-	L	-	-	-	-	-	-	-	H	-	-	-	-	-	-	-	0	.	.	.	.	.					
2	-	-	-	L	-	-	-	L	-	-	-	-	-	-	H	H	-	-	-	-	-	-	-	0	.	.	.	.	.					
3	-	-	-	L	-	-	-	L	-	-	-	-	-	-	H	H	H	-	-	-	-	-	-	0	.	.	.	.	.					
4	-	-	-	L	-	-	-	L	-	-	-	-	-	-	H	H	H	H	-	-	-	-	-	0	.	.	.	.	.					
5	-	-	-	L	-	-	-	L	-	-	-	-	-	-	H	H	H	H	H	-	-	-	-	0	.	.	.	.	.					
6	-	-	-	L	-	-	-	L	-	-	-	-	-	-	H	H	H	H	H	H	-	-	-	0	.	.	.	.	.					
7	-	-	-	L	-	-	-	L	-	-	-	-	-	-	H	H	H	H	H	H	H	-	-	0	.	.	.	.	.					
8	-	-	-	L	-	-	-	L	-	-	-	-	-	-	H	H	H	H	H	H	H	H	-	-	.	.	.	.	A					
9																																		
10																																		
11																																		
12																																		
13																																		
14																																		
15																																		
16																																		
17																																		
18																																		
19																																		
20																																		
21																																		
22																																		
23																																		
24																																		
25																																		
26																																		
27																																		
28																																		
29																																		
30																																		
31																																		
FC																																		
PB																																		
RB																																		
LB	-	-	-	-	-	-	-	-	-	H	H	H	H	H	H	H																		
PA																																		
RA																																		
LA	-	-	-	-	-	-	-	-	-	H	H	H	H	H	H	H																		
D3																																		
D2																																		
D1																																		
D0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-																		
PIN	5	4	3	2	9	8	7	6	19	18	17	16	15	14	13	12	19	18	17	16	15	14	13	12	9	8	7	6						
REMARKS																																		
ENABLE																																		
PLS159																																		

TB000435

# PLS159 As Modulo Counters

# AN16



- NOTES:**
1. All OR gate inputs with a blown link float to logic "0".
  2. All other gates and control inputs with a blown link float to logic "1".
  3. ⊕ denotes WIRE-OR.

Figure 2

LD018815

# AN17

## Electronic Combination Lock

### Application Note

#### Application Specific Products

#### INTRODUCTION

A combination lock may be opened by entering the correct numbers which match the preset security code that may be of any number of digits. Using a PLS157, an electronic combination lock with user settable security code may be constructed. The security code may be one to five hexadecimal digits that can be preset by the user. The lock takes a 4-bit binary input as the "combination" and a "Data Valid" input, which means that the combination may be five digits long and each digit may be from 0 to F. If the input matches the preset digit of the security code, the lock will wait for the next digit. If the input does not match the corresponding digit of the security code, the lock will reset itself to the first digit position again. The user may use as many as five digits for the combination which may be set by using five DIP switches or other means. The lock may be used in conjunction with a 4-bit binary encoded rotary switch, or an encoded push button key pad, or an optical card reader, or a magnetic strip reader. The design takes advantage of the six JK flip-flops of the PLS157, the bidirectional I/O's, the complement array, and the general architecture of common product terms.

#### DESCRIPTION

Figure 1 shows an example of an electronic lock which uses five DIP switches to set the security code (hexadecimal 0 to E, F is reserved for reset), a 4-bit binary encoded rotary switch to enter the combination at  $I_0$  to  $I_3$ , and a single pole double throw push button switch, "Data Valid", as the clock which is debounced by the associated circuitry of pin  $B_5$ . The security code is set with five DIP switches which are connected to a four line

input ( $B(I)_0$  to  $B(I)_3$ ). The five DIP switches are multiplexed by the outputs of the flip-flops,  $F_0$  to  $F_4$ . In normal operation, one of the F/F outputs is Low (the content of the register being a "1") while the other five are High. The operation is actually a shift register with a zero bit shifting through five one's. Therefore only one DIP switch is pulled Low (selected) at any one time while the rest is deselected. The pin named "MATCH",  $B(O)_4$ , is an I/O which outputs a High if the input matches the corresponding digit of the security code, otherwise it outputs a Low. A High output of  $B(O)_4$  enables the shift register to shift one position while a Low causes the shift register to reset to the starting position which is a "1" in F/F<sub>0</sub> (flip-flop 0) ( $F_0$  is Low due the output inverter) and "0's" in the other five ( $F_1$  to  $F_5$  = High). The state diagram of the design is as shown in Figure 2 where the state numbers are expressed in two hexadecimal digits. The least significant digit consists of  $F_5$ ,  $F_2$ ,  $F_1$ , and  $F_0$ ; whereas the most significant digit consists of  $F_5$ , and  $F_4$ . At the top of the state diagram, the power-up and reset state is 00 hex. When the outputs have more than one Low, as sometimes happens during power-up, the circuit automatically resets the registers to 00 hex which causes all outputs to be High in order to prevent more than one DIP switch being enabled. After power-up or reset, pushing the "Data Valid" button will initialize the circuit to have the first digit of the security code enabled. Subsequently, any false entry will cause the circuit to go back to the initialized state of 01 hex. When the first digit is correctly set and the "Data Valid" button pressed, the circuit goes to 02 hex and the second DIP switch is enabled while the first DIP switch is disabled. The circuit is now ready for the second digit. By repeating the process, the circuit keeps shifting the "1" in

the shift register until it reaches F/F<sub>5</sub> at which time the Low output may be used to trigger a solenoid. If at any time of the sequence, an incorrect digit is entered, the circuit reverts back to the initialized state of 01 hex.

Notice that any of the five outputs from  $F_1$  to  $F_5$  may be used as an output. Therefore, if, for example, a two digit security code is used, only two DIP switches are needed and the output is taken from  $F_2$  to trigger a solenoid. After the last digit has been entered, all DIP switches are deselected and the inputs  $B(I)_0$  to  $B(I)_3$  become 1111 binary, or F hex. Therefore, the hex number F is not used as a valid code but as an initialization input entered at  $I_0$  to  $I_3$ . When the hex number F is entered, the circuit initializes itself to the 01 hex state. After the last digit of the security code has been entered, if the "Data Valid" button is pressed again, the circuit sees a false entry and reinitializes itself.

The design is implemented using a PLS157 as shown in Table 1, H/L Programming Table of Electronic Lock. P-terms 0 to 14 are programmed as decoders. If an input matches the corresponding digit of the security code, one of the p-terms will be active which causes  $B(O)_4$  to output a High which, in turn, enables the registers to shift to the next digit by terms 16 to 20. In the event that an incorrect input is entered,  $B(O)_4$  outputs a Low which, in turn, causes the registers to be set 000001 binary by term 21. If an input of F is entered, term 15 causes the registers to be 000001. Terms 22 to 29 and RB ensure that no more than one register is "1" at a time. Terms 22 to 28 are connected to the complement array, which in turn, drives terms 29 and RB to set the registers to 000000 when terms 22 to 28 are inactive.



# Electronic Combination Lock

AN17

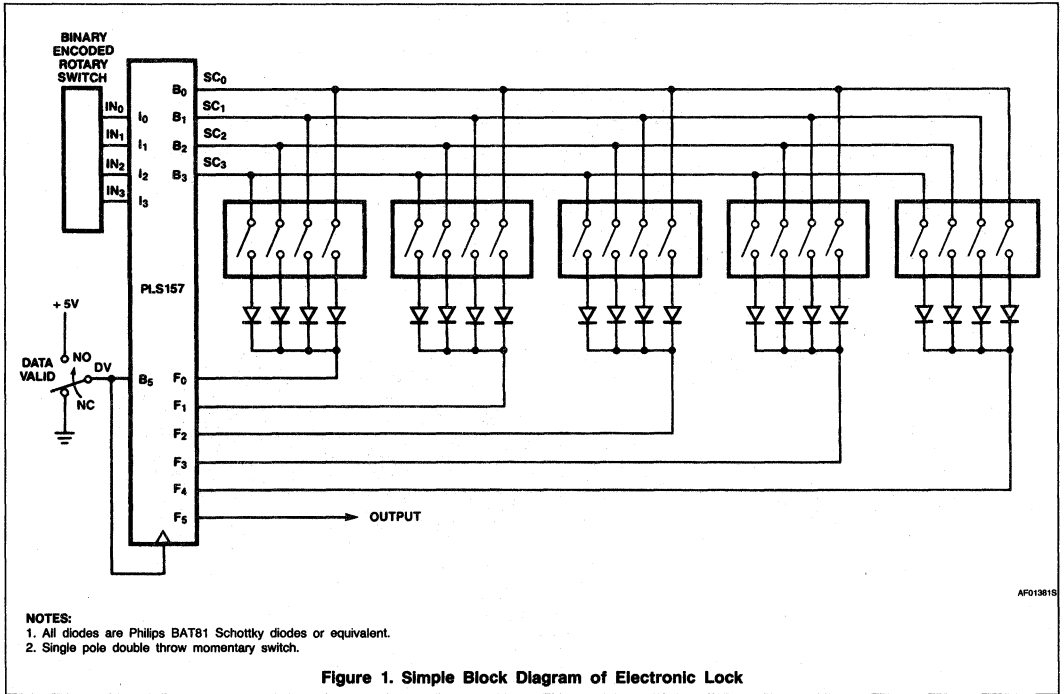


Figure 1. Simple Block Diagram of Electronic Lock

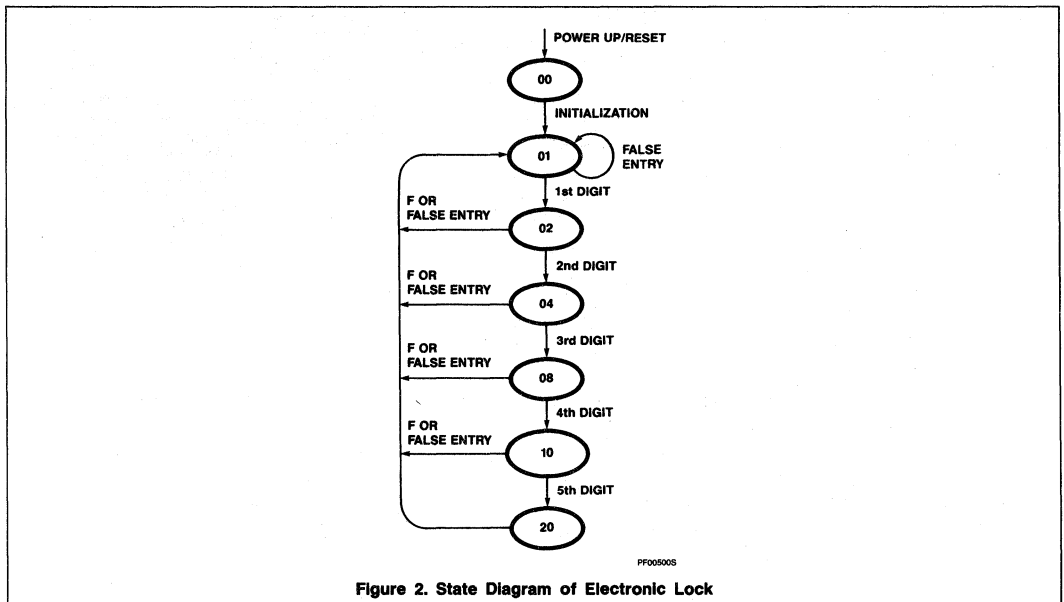


Figure 2. State Diagram of Electronic Lock

Electronic Combination Lock

AN17

Table 1. PLS157 H/L Programming Table

T*	F/F MODE										REMARKS	EA : EB		POLARITY										REMARKS					
	PMD											Q(N)	P:A	B(O)															
	I			B(I)			Q(P)								H			H			H								
C	3	2	1	0	5	4	3	2	1	0	5	4	3	2	1	0	A	A	5	4	3	2	1	0					
00	L	L	L	L	L	L	L	L	L	L	0						A	A								BH = L IE			
01	L	L	L	L	H	L	L	L	L	H	1						A	A								INPUT MATCHING			
02	L	L	L	H	L	L	L	L	H	L	2						A	A								5-BANK ON LOCKS?			
03	L	L	H	H	L	L	L	L	H	L	3						A	A											
04	L	H	L	L	L	L	L	L	H	L	4						A	A											
05	L	H	L	H	L	L	L	L	H	L	5						A	A											
06	L	H	H	L	L	L	L	L	H	L	6						A	A											
07	L	H	H	H	L	L	L	L	H	L	7						A	A											
08	H	L	L	L	L	L	L	L	H	L	8						A	A											
09	H	L	L	H	L	L	L	L	H	L	9						A	A											
10	H	L	H	L	L	L	L	L	H	L	A						A	A											
11	H	L	H	H	L	L	L	L	H	L	B						A	A											
12	H	H	L	L	L	L	L	L	H	L	C						A	A											
13	H	H	L	H	L	L	L	L	H	L	D						A	A											
14	H	H	H	L	L	L	L	L	H	L	E						A	A											
15	H	H	H	H	L	L	L	L	H	L	F						A	A											
16					H	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	INITIALIZE TO PROGRAM		
17					H	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	SHIFT RIGHT		
18					H	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L			
19					H	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L			
20					H	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L			
21					L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	INITIALIZE		
22	A										L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	INITIALIZE		
23	A										L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	IF ALL ARE 0		
24	A										L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	OR STOP PROGRAM?		
25	A										L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	F/F = 2		
26	A										L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L			
27	A										L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L			
28	A										H	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L			
29	*																A	A									IC = 2		
30																											RESET F <sub>1</sub> -F <sub>2</sub>		
31					H												A	A									"DY" DEBOUNCE		
FC																											ALL F/F = JK		
PB																													
RB	*																										IC = 1		
LB																											RESET F/F = JK		
LA																													
D5																											ENABLE B01E		
D4																											ENABLE B01H		
D3																													
D2																													
D1																													
D0																													
PIN	*	5	4	3	2	19	12	9	8	7	6	18	17	16	15	14	13	18	17	16	15	14	13	19	12	9	8	7	6
N																													
R																													
H																													
M																													
E																													

T8004305

#### Application Specific Products

#### INTRODUCTION

One of the many features of the PLS153 to 159 series is the availability of individually controlled Tri-state I/O pins. Taking advantage of this feature, a Schmitt trigger may be constructed using one input pin, two bidirectional I/O pins and additional components of three resistors. The two threshold voltages, as well as the hysteresis, are determined by the values of the three resistors and the parameters of the PLS153/159 device, which are 1) input threshold voltage,  $V_{TH}$ , 2) High output voltage,  $V_{OH}$ , and 3) Low output voltage,  $V_{OL}$ . The circuit may be simplified if Schmitt function is needed only on Low going High or High going Low, and if the hysteresis and threshold voltages are not important.

#### DESCRIPTION

A simplified block diagram of a non-inverting Schmitt trigger is shown in Figure 1 where  $R_1$ ,  $R_2$ , and  $R_3$  form two pairs of voltage dividers one of which get into action at input voltage direction of High going Low and the other Low going High. Assuming that input voltage starts at zero volt, the output voltage is therefore at  $V_{OL}$  which causes  $Q_2$  to pull  $R_3$  towards ground. As the input voltage increases, only a fraction of the voltage is

impressed upon the input buffer due to the dividing network  $R_1$  and  $R_3$ . As soon as the input voltage reaches a point where  $V_1 = V_{TH}$  ( $V_{TH} = 1.38V$  typical), the output switches to  $V_{OH}$  which, in turn, turns off  $Q_2$  and turns on  $Q_1$ .  $V_1$  will jump to a value greater than  $V_{TH}$  and  $Q_1$  then pulls the input pin, through  $R_2$ , towards  $V_{OH}$ , which in turn locks the output to a High state even if the input voltage fluctuates, as long as it does not fluctuate outside of the designed hysteresis. When the input voltage goes from a High to a Low, the Schmitt function repeats itself except that  $Q_1$  and  $Q_2$  reverse their roles.

The triggering voltages,  $V_H$  (Low going High) and  $V_L$  (High going Low) are:

$$V_H = V_{TH} [(R_1 + R_3)/R_3] - V_{OL} (R_1/R_3);$$

$$V_L = V_{TH} [(R_1 + R_2)/R_2] - V_{OH} (R_1/R_2);$$

where, at room temperature,  $V_{CC} = 5.0V$ ,  $I_{OH}/I_{OL} < 1mA$ .  $V_{TH}$  is the threshold voltage of the device, typically 1.38V;  $V_{OL}$  is the output Low voltage of the device, typically 0.36V at  $|I_{OL}| < 1mA$ ;  $V_{OH}$  is the output High voltage of the device, typically 3.8V at  $|I_{OH}| < 1mA$ .

The implementation of Figure 1 using PLS153/153A is as shown in Table 1, and Figure 2a. A scope photo of the operation of the circuit is shown in the Appendix. The

implementation using PLS159 is shown in Table 2 and Figure 2b. In Tables 1 & 2,  $V_1$  is the input pin,  $V_0$  is the output pin,  $V_2$  is the output which pulls down  $V_1$  and  $V_3$  is the output pin that pulls up  $V_1$ . The Schmitt output is available at pin  $B_0$  for external use, and is available internally at the input buffers of  $I_0$  and  $B(I)_0$ . However, there is a propagation delay between the two signals from the  $I_0$  buffer and the  $B(I)_0$  buffer.

An inverting Schmitt triggered buffer may be constructed using the same principle. A simple block diagram of such inverter is shown in Figure 3a. The circuit is implemented using H/L programming table as shown in Table 3 for PLS153 and Table 4 for PLS159. Table 3 is also represented in logic symbols in Figure 3b. If the voltage levels ( $V_L$  and  $V_H$ ) and the hysteresis are not critical, one I/O pin may be used to pull the input pin High and Low. Therefore one I/O pin and a resistor may be saved. The drawback is that the range of  $V_H$  and  $V_L$  is quite limited. The circuit is as shown in Figure 4.

If Schmitt function is needed only in one direction, one of the resistor/output circuit may be eliminated. The circuit is as shown in Figure 5.

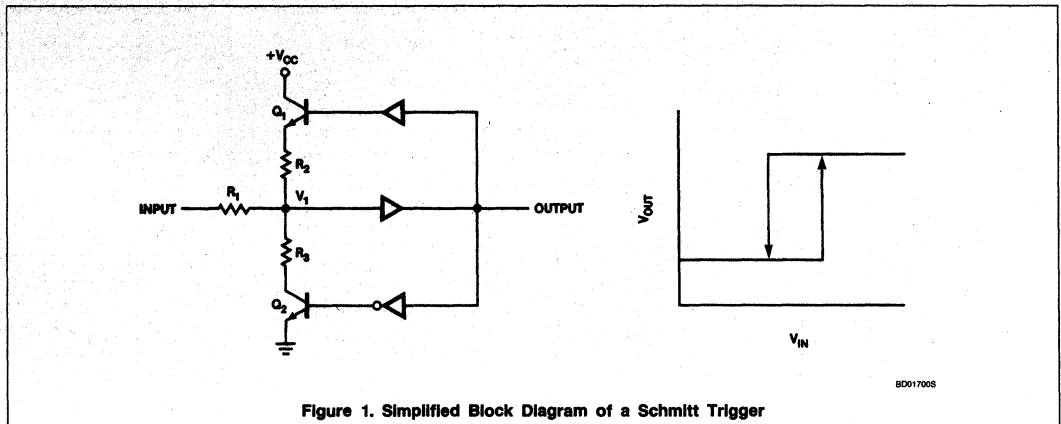


Figure 1. Simplified Block Diagram of a Schmitt Trigger

# Schmitt Trigger Using PLS153 and PLS159

AN18

Table 1. PLS153/153A Programming Table

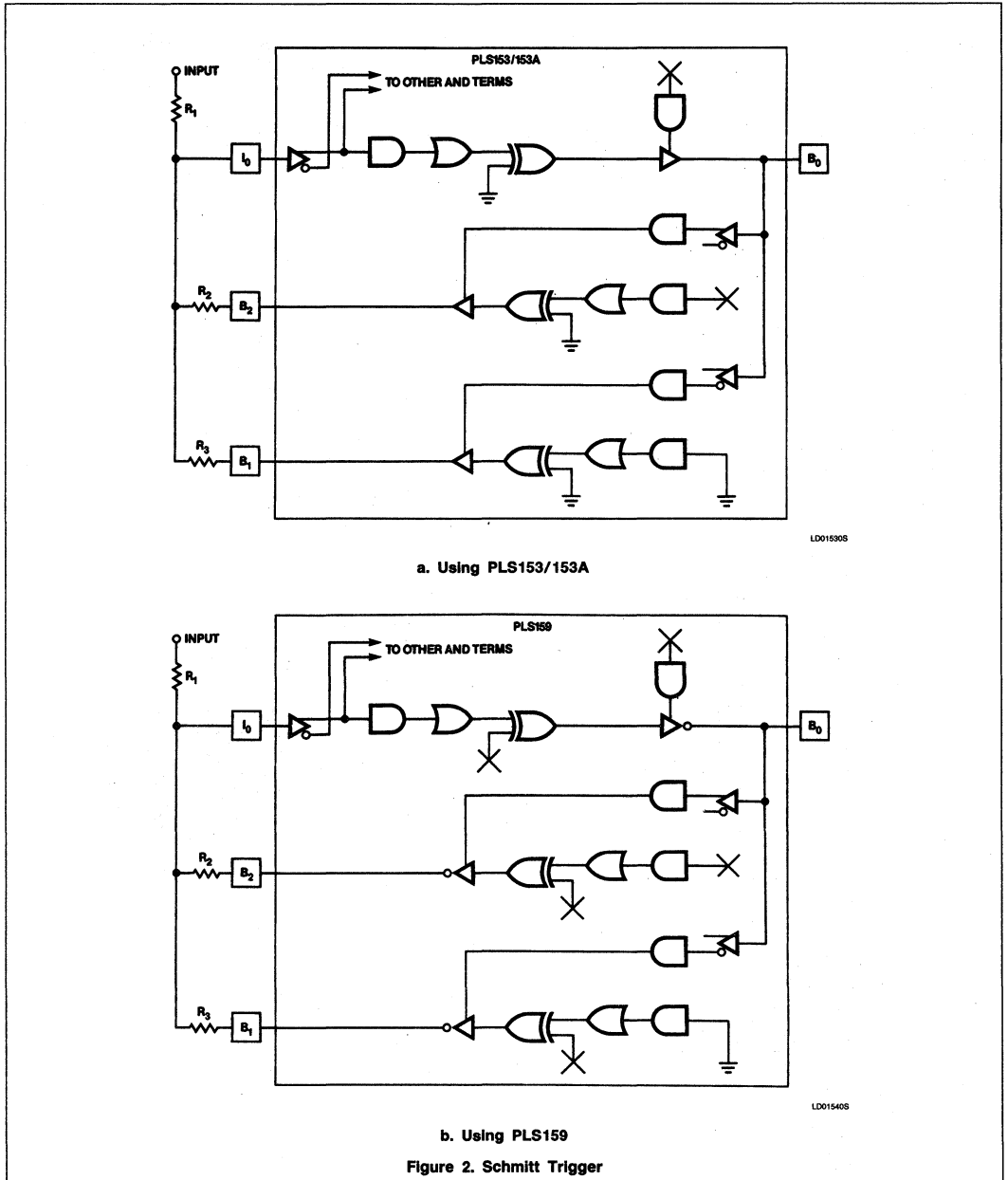
CODE NO.													POLARITY																	
TERM	AND												REMARKS	L	H	H	H	OR						REMARKS						
	I						B(I)							B(O)																
	7	6	5	4	3	2	1	0	9	8	7	6		5	4	3	2	1	0	9	8	7	6		5	4	3	2	1	0
0	-	-	-	-	-	-	H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A	NON-INV. BUFFER	
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	A	OUTPUT V <sub>OL</sub>		
2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A	OUTPUT V <sub>OH</sub>		
3																														
4																														
5																														
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	A	SAME AS TERMS 1+2		
7																											WITH			
8																											B(0)4 = V <sub>2</sub>			
9																											B(0)5 = V <sub>2</sub>			
10																														
11																														
12																														
13																														
14																														
15																														
16																														
17																														
18																														
19																														
20																														
21																														
22																														
23																														
24																														
25																														
26																														
27																														
28																														
29																														
30																														
31																														
D9																														
D8																														
D7																														
D6																														
D5																														
D4																														
D3																														
D2																										H				
D1																										L				
D0																														
PN	8	7	6	5	4	3	2	1	19	18	17	16	15	14	13	12	11	9	19	18	17	16	15	14	13	12	11	9		
REMARKS								V <sub>1</sub>																					V <sub>2</sub> V <sub>2</sub> V <sub>2</sub>	PLS153

NOTE:  
Schmitt trigger output may be obtained from both 1<sub>0</sub> and B(1)<sub>0</sub> to drive the AND-ARRAY.

T8017305

# Schmitt Trigger Using PLS153 and PLS159

AN18



# Schmitt Trigger Using PLS153 and PLS159

AN18

Table 2. PLS159 FPLS Programming Table

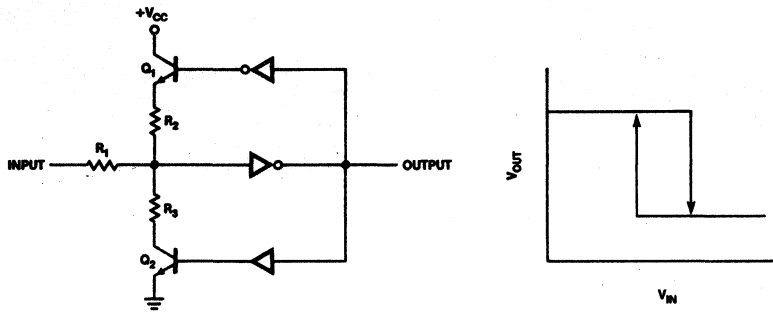
CODE NO.		FF MODE											E <sub>B</sub>		E <sub>A</sub>		POLARITY			REMARKS										
T E R M	C	AND						REMARKS	(OR)												REMARKS									
		I			B(I)				Q(P)						Q(N)							B(O)								
		3	2	1	0	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	3	2	1	0	
0	-	-	-	-	H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	•	•	•	A	NON-INV. BUFR
1	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	A	OUTPUT V <sub>OL</sub>
2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	•	A	•	•	OUTPUT V <sub>OH</sub>
3																														
4																														
5																														
6																														
7																														
8																														
9																														
10																														
11																														
12																														
13																														
14																														
15																														
16																														
17																														
18																														
19																														
20																														
21																														
22																														
23																														
24																														
25																														
26																														
27																														
28																														
29																														
30																														
31																														
FC																														
PB																														
RB																														
LB																														
PA																														
RA																														
LA																														
D3																														
D2	-	-	-	-	-	-	-	-	H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
D1	-	-	-	-	-	-	-	-	L	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
D0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
PIN		5	4	3	2	9	8	7	6	19	18	17	16	15	14	13	12	19	18	17	16	15	14	13	12	9	8	7	6	
REMARKS						V <sub>1</sub>																								
							V <sub>2</sub>																							
								V <sub>2</sub>																						
									V <sub>6</sub>																					

PLS159

TB017405

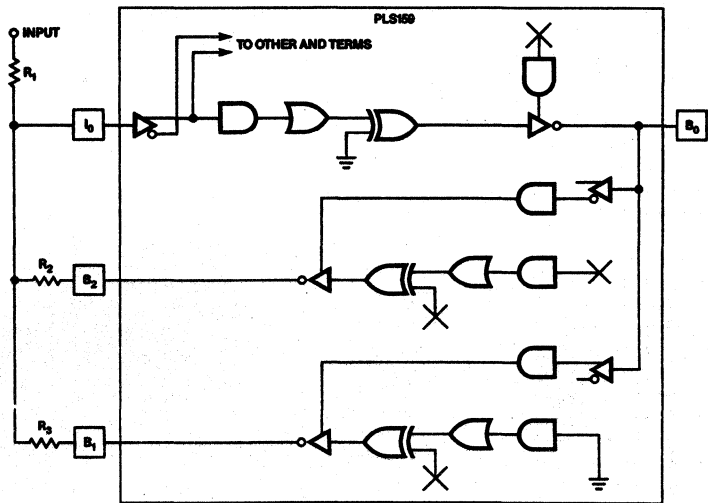
# Schmitt Trigger Using PLS153 and PLS159

AN18



80017105

a. Simplified Block Diagram



LD015505

b. Using PLS159

Figure 3. Inverting Schmitt Trigger

# Schmitt Trigger Using PLS153 and PLS159

AN18

Table 3. PLS153/153A Programming Table

CODE NO.														POLARITY										REMARKS						
T	AND										OR																			
	B(I)										B(O)																			
M	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	REMARKS	9	8	7	6	5	4	3	2	1	0	REMARKS
0	-	-	-	-	-	-	-	H	-	-	-	-	-	-	-	-	-	-											• A INV. BUFFER	
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0											• A OUTPUT V <sub>OH</sub>		
2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-											A • OUTPUT V <sub>OL</sub>		
3																														
4																														
5																														
6																														
7																														
8																														
9																														
10																														
11																														
12																														
13																														
14																														
15																														
16																														
17																														
18																														
19																														
20																														
21																														
22																														
23																														
24																														
25																														
26																														
27																														
28																														
29																														
30																														
31																														
D9																														
D8																														
D7																														
D6																														
D5																														
D4																														
D3																														
D2																												H		
D1																												L		
D0																														
PN	8	7	8	5	4	3	2	1	19	18	17	16	15	14	13	12	11	9												
REMARKS								V <sub>i</sub>																				V <sub>o</sub> V <sub>e</sub> V <sub>i</sub>		

PLS153

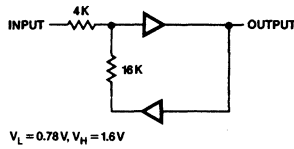
TB010005





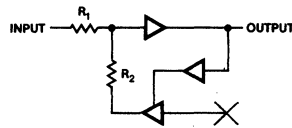
# Schmitt Trigger Using PLS153 and PLS159

AN18



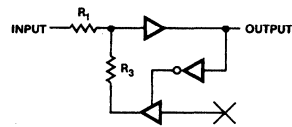
TC01890S

Figure 4. Schmitt Trigger Using One I/O Pin



TC01900S

a. High Going Low Direction

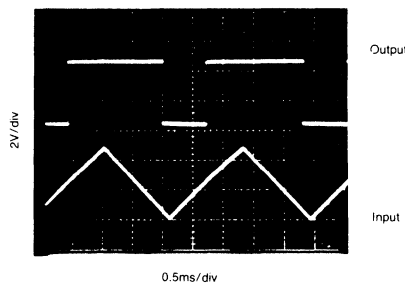


TC01910S

b. Low Going High Direction

Figure 5. Schmitt Trigger

## APPENDIX A



NOTE:  
 $R_1 = 3.9k\Omega$ ,  $R_2 = 10.8k\Omega$ ,  $R_3 = 2.0k\Omega$ ,  $V_{CC} = 5.0V$ , Ambient temperature  $\approx 25^\circ C$

Figure A-1. A Non-Inverting Schmitt Triggered Buffer

### Application Specific Products

#### DESCRIPTION

A PLS159 is used to implement a pair of dice. Each die is represented by a series of six different state transitions. Two push button switches are used to "roll" the dice; in other words start/stop the state transitions. Each push button controls one die. When the switches are held low, the state transitions take place for each die and the result is displayed when a push button is released.

Each die is made up of seven LED's which are connected to the outputs of the PLS159. The system also contains its own oscillator which generates the clock cycle necessary for the state transitions.

The random release of the push button provides a random output. The high frequency of the internal oscillator and the fact that the switches are not debounced, add more "randomness" to each event.

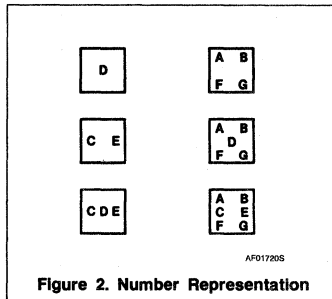
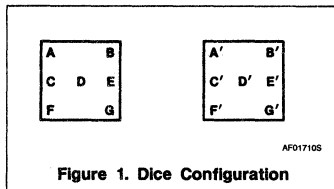
The overall system configuration is shown in Figure 3.

#### LOGIC IMPLEMENTATION

The configuration of each die is shown in Figure 1, with each letter representing an LED.

To represent the numbers one through six the LED's are turned on as shown in Figure 2.

Table 1 is the truth table generated from Figure 2.



**Table 1. Truth Table for Dice**

	A	B	C	D	E	F	G
One	0	0	0	1	0	0	0
Two	0	0	1	0	1	0	0
Three	0	0	1	1	1	0	0
Four	1	1	0	0	0	1	1
Five	1	1	0	1	0	1	1
Six	1	1	1	0	1	1	1

The result of Table 1 is the following logic equations:

$$\begin{aligned}
 A &= \text{Four} + \text{Five} + \text{Six}; \\
 B &= \text{Four} + \text{Five} + \text{Six}; \\
 C &= \text{Two} + \text{Three} + \text{Six}; \\
 D &= \text{One} + \text{Three} + \text{Five}; \\
 E &= \text{Two} + \text{Three} + \text{Six}; \\
 F &= \text{Four} + \text{Five} + \text{Six}; \\
 G &= \text{Four} + \text{Five} + \text{Six};
 \end{aligned}$$

As can be seen from the above logic equations:

$$\begin{aligned}
 A &\equiv B \equiv F \equiv G; \\
 C &\equiv E;
 \end{aligned}$$

Table 2 is the result of Table 1 and the above common terms.

**Table 2. Truth Table for PLS159 Outputs**

	A, B, F, G	C, E	D
One	0	0	1
Two	0	1	0
Three	0	1	1
Four	1	0	0
Five	1	0	1
Six	1	1	0

For each die, three outputs of the PLS159 are needed to implement the logic in Table 2. These outputs are connected to the LED's as follows:

$$\begin{aligned}
 F0 &: A, B, F, G; \\
 F1 &: C, E; \\
 F2 &: D;
 \end{aligned}$$

Similarly for the second die:

$$\begin{aligned}
 F3 &: A', B', F', G'; \\
 F4 &: C', E'; \\
 F5 &: D';
 \end{aligned}$$

Electronic Dice

AN19

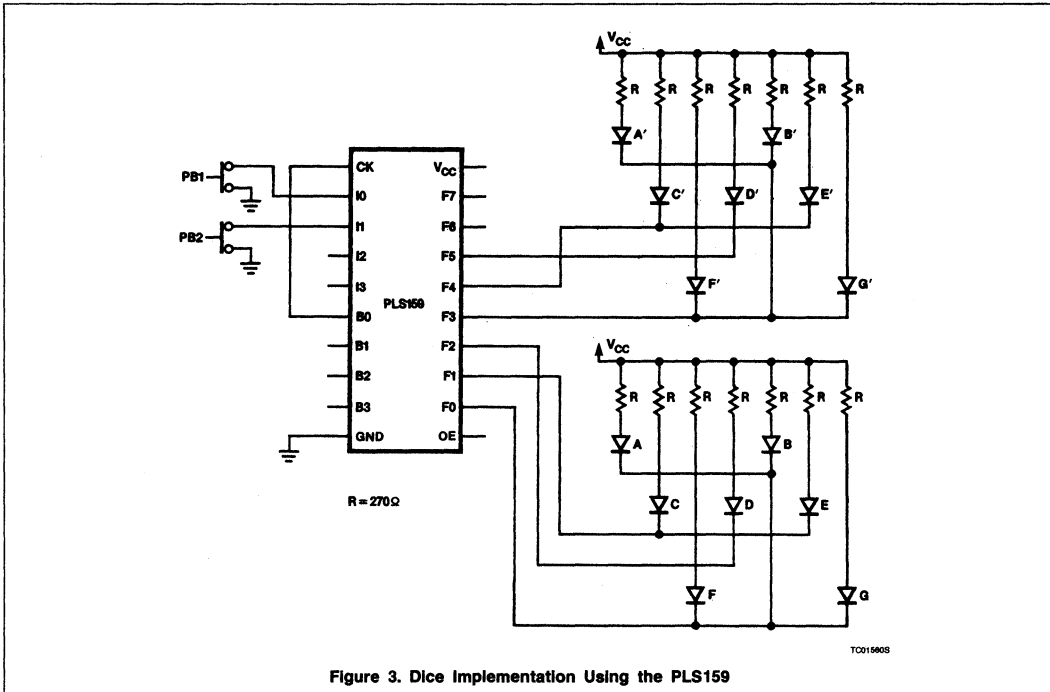


Figure 3. Dice Implementation Using the PLS159

# Electronic Dice

# AN19

### STATE TRANSITIONS

Figure 4 shows two arbitrary state transition sequences which are used to implement each die.

These state transitions are implemented in the program table of the PLS159.

### INTERNAL OSCILLATOR

An internal oscillator is also implemented for the system. The internal configuration of the oscillator is shown in Figure 5. For a detailed explanation of the implementation of this oscillator see AN13: Oscillator with the PLS159. B0 is connected externally to the clock input of the PLS159.

### EXPLANATION OF PROGRAM TABLE

The program table for the two dice is shown in Table 3.

Terms 0 - 5 and 7 - 12 represent the state transitions for each die.

Terms 14 - 15 and 17 - 18 reset the flip-flops to HLL and HLH if undefined states occur, such as during power up.

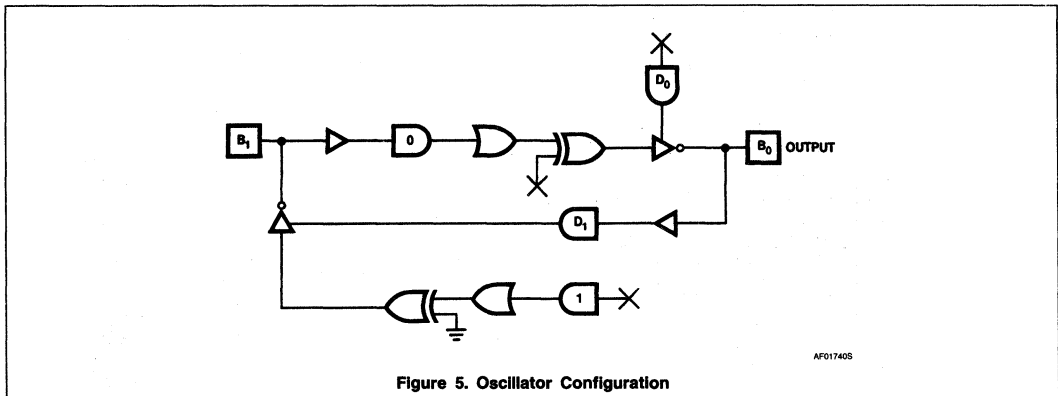
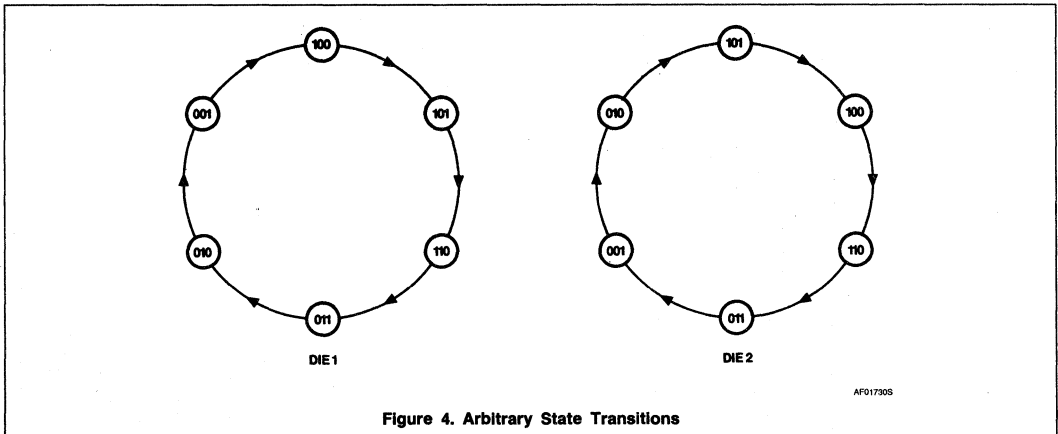
Terms 20 and 21 and output controls terms D1 and D2 are used to implement the internal oscillator (see AN13).

Inputs I0 and I1 are connected to the two push button switches which roll the dice. These inputs are programmed to be Active-Low. When the push button is closed the state transitions take place, and upon release of the push button, the current states are displayed on the outputs.

The outputs of the PLS159 are inverted and this must be taken into consideration when assigning "H" and "L" to the F0, F1, F2, F3, F4, and F5. These assignments are shown in Table 4. All the flip-flops are set to be of the J - K type.

**Table 4. H And L Assignment**

	F2, F5	F1, F4	F0, F3
One	H	L	L
Two	L	H	L
Three	H	H	L
Four	L	L	H
Five	H	L	H
Six	L	H	H



Electronic Dice

AN19

Table 3. Program Table for 2 Dice

CODE NO.		FF MODE										REMARKS	E <sub>B</sub>				E <sub>A</sub>				POLARITY				REMARKS			
TERM	C	AND											REMARKS	(OR)														
		I			B(I)			Q(P)				Q(N)				B(O)												
		3	2	1	0	3	2	1	0	7	6	5		4	3	2	1	0	7	6	5	4	3	2	1	0	3	2
0	-	-	-	L	-	-	-	-	-	-	-	-	H	L	L	100 → 101	-	-	-	-	H	L	H	•	•	•	•	
1	-	-	-	L	-	-	-	-	-	-	-	-	H	L	H	101 → 110	-	-	-	-	H	H	L	•	•	•	•	
2	-	-	-	L	-	-	-	-	-	-	-	-	H	H	L	110 → 011	-	-	-	-	L	H	H	•	•	•	•	
3	-	-	-	L	-	-	-	-	-	-	-	-	L	H	H	011 → 010	-	-	-	-	L	H	L	•	•	•	•	
4	-	-	-	L	-	-	-	-	-	-	-	-	L	H	L	010 → 001	-	-	-	-	L	L	H	•	•	•	•	
5	-	-	-	L	-	-	-	-	-	-	-	-	L	L	H	001 → 100	-	-	-	-	H	L	L	•	•	•	•	
6	-	-	-	L	-	-	-	-	-	-	-	-	-	-	-	101 → 100	-	-	H	L	L	-	-	•	•	•	•	
7	-	-	-	L	-	-	-	-	-	-	-	-	H	L	L	100 → 110	-	-	H	H	L	-	-	•	•	•	•	
8	-	-	-	L	-	-	-	-	-	-	-	-	H	H	L	110 → 011	-	-	L	H	H	-	-	•	•	•	•	
9	-	-	-	L	-	-	-	-	-	-	-	-	L	H	H	011 → 001	-	-	L	L	H	-	-	•	•	•	•	
10	-	-	-	L	-	-	-	-	-	-	-	-	L	L	H	001 → 010	-	-	L	H	L	-	-	•	•	•	•	
11	-	-	-	L	-	-	-	-	-	-	-	-	L	H	L	010 → 101	-	-	H	L	H	-	-	•	•	•	•	
12	-	-	-	L	-	-	-	-	-	-	-	-	-	-	-	Undefined states for Die 1	-	-	-	-	H	L	L	•	•	•	•	
13	-	-	-	L	-	-	-	-	-	-	-	-	-	-	-	Undefined states for Die 2	-	-	-	-	H	L	L	•	•	•	•	
14	-	-	-	L	-	-	-	-	-	-	-	-	L	L	L	Go to 100	-	-	-	-	H	L	L	•	•	•	•	
15	-	-	-	L	-	-	-	-	-	-	-	-	H	H	H	Go to 101	-	-	-	-	H	L	H	•	•	•	•	
16	-	-	-	L	-	-	-	-	-	-	-	-	L	L	L	Go to 101	-	-	-	-	H	L	H	•	•	•	•	
17	-	-	-	L	-	-	-	-	-	-	-	-	H	H	H	Go to 101	-	-	-	-	H	L	H	•	•	•	•	
18	-	-	-	L	-	-	-	-	-	-	-	-	-	-	-	Oscillator	-	-	-	-	•	•	•	•	A	•	E <sub>B</sub> = B <sub>1</sub>	
19	-	-	-	L	-	-	-	-	-	-	-	-	-	-	-	Oscillator	-	-	-	-	•	•	•	•	A	•	B(O) <sub>1</sub> = Low	
20	-	-	-	L	-	-	-	-	-	-	-	-	-	-	-	Oscillator	-	-	-	-	•	•	•	•	A	•	B(O) <sub>1</sub> = Low	
21	-	-	-	L	-	-	-	-	-	-	-	-	-	-	-	Oscillator	-	-	-	-	•	•	•	•	A	•	B(O) <sub>1</sub> = Low	
22	-	-	-	L	-	-	-	-	-	-	-	-	-	-	-	Oscillator	-	-	-	-	•	•	•	•	A	•	B(O) <sub>1</sub> = Low	
23	-	-	-	L	-	-	-	-	-	-	-	-	-	-	-	Oscillator	-	-	-	-	•	•	•	•	A	•	B(O) <sub>1</sub> = Low	
24	-	-	-	L	-	-	-	-	-	-	-	-	-	-	-	Oscillator	-	-	-	-	•	•	•	•	A	•	B(O) <sub>1</sub> = Low	
25	-	-	-	L	-	-	-	-	-	-	-	-	-	-	-	Oscillator	-	-	-	-	•	•	•	•	A	•	B(O) <sub>1</sub> = Low	
26	-	-	-	L	-	-	-	-	-	-	-	-	-	-	-	Oscillator	-	-	-	-	•	•	•	•	A	•	B(O) <sub>1</sub> = Low	
27	-	-	-	L	-	-	-	-	-	-	-	-	-	-	-	Oscillator	-	-	-	-	•	•	•	•	A	•	B(O) <sub>1</sub> = Low	
28	-	-	-	L	-	-	-	-	-	-	-	-	-	-	-	Oscillator	-	-	-	-	•	•	•	•	A	•	B(O) <sub>1</sub> = Low	
29	-	-	-	L	-	-	-	-	-	-	-	-	-	-	-	Oscillator	-	-	-	-	•	•	•	•	A	•	B(O) <sub>1</sub> = Low	
30	-	-	-	L	-	-	-	-	-	-	-	-	-	-	-	Oscillator	-	-	-	-	•	•	•	•	A	•	B(O) <sub>1</sub> = Low	
31	-	-	-	L	-	-	-	-	-	-	-	-	-	-	-	Oscillator	-	-	-	-	•	•	•	•	A	•	B(O) <sub>1</sub> = Low	
FC	-	-	-	L	-	-	-	-	-	-	-	-	-	-	-	Oscillator	-	-	-	-	•	•	•	•	A	•	B(O) <sub>1</sub> = Low	
P <sub>1</sub>	-	-	-	L	-	-	-	-	-	-	-	-	-	-	-	Oscillator	-	-	-	-	•	•	•	•	A	•	B(O) <sub>1</sub> = Low	
RB	-	-	-	L	-	-	-	-	-	-	-	-	-	-	-	Oscillator	-	-	-	-	•	•	•	•	A	•	B(O) <sub>1</sub> = Low	
LB	-	-	-	L	-	-	-	-	-	-	-	-	-	-	-	Oscillator	-	-	-	-	•	•	•	•	A	•	B(O) <sub>1</sub> = Low	
PA	-	-	-	L	-	-	-	-	-	-	-	-	-	-	-	Oscillator	-	-	-	-	•	•	•	•	A	•	B(O) <sub>1</sub> = Low	
RA	-	-	-	L	-	-	-	-	-	-	-	-	-	-	-	Oscillator	-	-	-	-	•	•	•	•	A	•	B(O) <sub>1</sub> = Low	
LA	-	-	-	L	-	-	-	-	-	-	-	-	-	-	-	Oscillator	-	-	-	-	•	•	•	•	A	•	B(O) <sub>1</sub> = Low	
DS	-	-	-	L	-	-	-	-	-	-	-	-	-	-	-	Oscillator	-	-	-	-	•	•	•	•	A	•	B(O) <sub>1</sub> = Low	
D2	-	-	-	L	-	-	-	-	-	-	-	-	-	-	-	Oscillator	-	-	-	-	•	•	•	•	A	•	B(O) <sub>1</sub> = Low	
D1	-	-	-	L	-	-	-	-	-	-	-	-	-	-	-	Oscillator	-	-	-	-	•	•	•	•	A	•	B(O) <sub>1</sub> = Low	
D0	-	-	-	L	-	-	-	-	-	-	-	-	-	-	-	Oscillator	-	-	-	-	•	•	•	•	A	•	B(O) <sub>1</sub> = Low	
PN	5	4	3	2	9	8	7	6	19	18	17	16	15	14	13	12	19	18	17	16	15	14	13	12	9	8	7	6
REMARKS																												

PLS159

#### Application Specific Products

#### INTRODUCTION

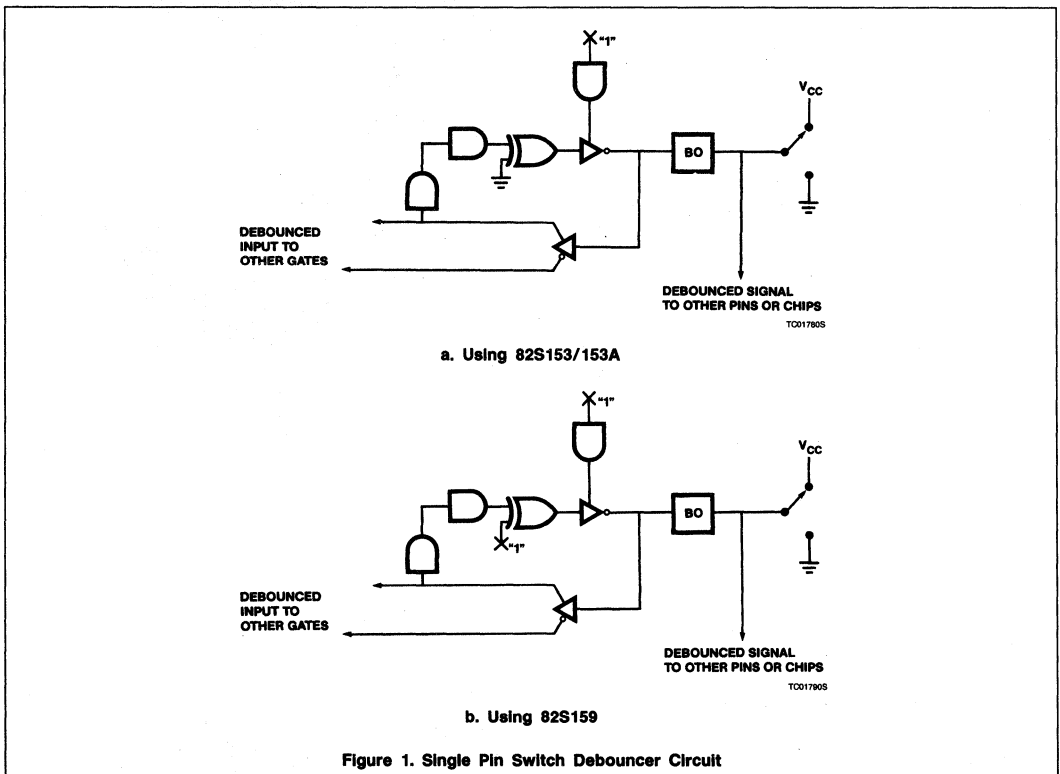
The 82S159 is a field-programmable logic sequencer which is self-contained as a logic sequencer except for the need of a system clock and/or one or more debounced switches. Generally, most of the debouncers circuits utilize a cross-coupled NAND/NOR to form an RS-latch which takes two input pins. To economize the I/O pin consumption, a single I/O pin may be used to debounce a single pole double throw switch. This scheme is applicable to all Signetics PLD's (Programmable Logic Devices) that have non-registered bi-directional I/O's, including 82S153/153A, and 82S155.

#### DESCRIPTION

Figures 1a and 1b are logic diagrams of the circuit which utilizes one I/O pin of the 82S153/153A and 82S159 respectively to form a debouncer. The single pole double throw switch is connected, at one terminal, to  $V_{CC}$ , and, at another terminal, to ground. The blade of the switch is connected to B0. B0 is programmed as an output. The input buffer of B0 is programmed to drive a p-term which, in turn, drives the output B0. The circuit latches to whichever state it is in. As an example, assuming that the output B0 is connected to  $V_{CC}$  through the switch, and its output is a "1". When the switch is flipped from  $V_{CC}$  to ground, B0 is momentarily short circuited to

ground. After one propagation delay (15-30ns), B0 is set to "0" and no more current will flow from B0. If the switch is flipped back to  $V_{CC}$ , B0 will be shorted to  $V_{CC}$  for one prop delay and will be set to a "1" and will stay a "1" until the switch is flipped again. The switch may bounce while the blade breaks from one terminal and bounce again while it makes contact with the other terminal. As long as the movement of the blade is smaller than the gap which separates the two terminals, the output will remain stable.

The circuit is implemented in an 82S159 using AMAZE as shown in Figures 2 & 3, and Table 1.



## Single Pin Debouncer Using 82S153/159

AN20

```

***** DEBOUNCE *****
***** P I N   L I S T *****

      LABEL      ** FNC **PIN ----- PIN** FNC ** LABEL
N/C      ** CK ** 1-!      !-20 ** +5V **VCC
N/C      ** I  ** 2-!      !-19 ** B  **N/C
N/C      ** I  ** 3-!      !-18 ** B  **N/C
N/C      ** I  ** 4-!      8  !-17 ** B  **N/C
N/C      ** I  ** 5-!      2  !-16 ** B  **N/C
SWITCH   ** O  ** 6-!      S  !-15 ** O  **INPUT_LATCH
N/C      ** /B ** 7-!      1  !-14 ** B  **N/C
N/C      ** /B ** 8-!      5  !-13 ** B  **N/C
N/C      ** /B ** 9-!      9  !-12 ** B  **N/C
GND      ** OV ** 10-!     !-11 ** /OE **N/C

```

TB005605

Figure 2. Pin List

```

***** DEBOUNCE *****
@DEVICE TYPE
82S159
@DRAWING
***** SINGLE PIN DEBOUNCER
@REVISION
***** REV -
@DATE
***** xx/xx/xxxx
@SYMBOL
***** DEBOUNCE
@COMPANY
***** SIGNETICS
@NAME
@DESCRIPTION
This circuit accepts a single pole double pole switch which switches between
Vcc and GROUND. BO is used as input, but it is defined in AMAZE as output.
BO may be used to drive the rest of the logic circuit or it may be used
in conjunction with one of the internal flip flops as registered input to
handle asynchronous input to a state machine.
@COMMON PRODUCT TERM
@COMPLEMENT ARRAY
@I/O DIRECTION
@OUTPUT POLARITY
@FLIP FLOP CONTROL
@OUTPUT ENABLE
@REGISTER LOAD
@ASYNCHRONOUS PRESET/RESET
@FLIP FLOP MODE
@LOGIC EQUATION
SWITCH = SWITCH ;

" *****
*          BO WITH LATCHED INPUT          *
***** "
/INPUT_LATCH : D = SWITCH ;

```

TB005605

Figure 3. Boolean Equation File



# Single Pin Debouncer Using 82S153/159

AN20

Table 1. Programming Table

```

***** DEBOUNCE *****
Cust/Project -
Date - ***** xx/xx/xxxx
Rev/I. D. - ***** REV -

B2S159          ! F/F TYPE          ! E(b)= !E(a) = !POLARTY!
-----
T !              !A!A!A!A!A!A!A!A!  0  !  0  !L!L!L!H!
E |
R ! ! I ! B(i) ! Q(p) ! Q(n) ! B(o) !
M !C!
-----
! !3_2_1_0!3_2_1_0!7_6_5_4_3_2_1_0!7_6_5_4_3_2_1_0!3_2_1_0!
0!A!- - -!- - -!H!- - -!- - -!0 0 0 0,A 0 0 0!A A A!
1!0!0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!0 0 0 0,0 0 0!A A A!
2!0!0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!0 0 0 0,0 0 0!A A A!
3!0!0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!0 0 0 0,0 0 0!A A A!
4!0!0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!0 0 0 0,0 0 0!A A A!
5!0!0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!0 0 0 0,0 0 0!A A A!
6!0!0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!0 0 0 0,0 0 0!A A A!
7!0!0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!0 0 0 0,0 0 0!A A A!
8!0!0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!0 0 0 0,0 0 0!A A A!
9!0!0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!0 0 0 0,0 0 0!A A A!
10!0!0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!0 0 0 0,0 0 0!A A A!
11!0!0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!0 0 0 0,0 0 0!A A A!
12!0!0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!0 0 0 0,0 0 0!A A A!
13!0!0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!0 0 0 0,0 0 0!A A A!
14!0!0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!0 0 0 0,0 0 0!A A A!
15!0!0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!0 0 0 0,0 0 0!A A A!
16!0!0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!0 0 0 0,0 0 0!A A A!
17!0!0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!0 0 0 0,0 0 0!A A A!
18!0!0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!0 0 0 0,0 0 0!A A A!
19!0!0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!0 0 0 0,0 0 0!A A A!
20!0!0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!0 0 0 0,0 0 0!A A A!
21!0!0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!0 0 0 0,0 0 0!A A A!
22!0!0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!0 0 0 0,0 0 0!A A A!
23!0!0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!0 0 0 0,0 0 0!A A A!
24!0!0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!0 0 0 0,0 0 0!A A A!
25!0!0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!0 0 0 0,0 0 0!A A A!
26!0!0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!0 0 0 0,0 0 0!A A A!
27!0!0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!0 0 0 0,0 0 0!A A A!
28!0!0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!0 0 0 0,0 0 0!A A A!
29!0!0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!0 0 0 0,0 0 0!A A A!
30!0!0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!0 0 0 0,0 0 0!A A A!
31!0!0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!0 0 0 0,0 0 0!A A A!
Fc!0!0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!
Pb! !0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!
Rb! !0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!
Lb! !0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!
Pa! !0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!
Ra! !0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!
La! !0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!
D3! !0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!
D2! !0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!
D1! !0 0 0!0 0 0!0 0 0!0 0 0,0 0 0 0!
DO! !- - -!- - -!- - -!- - -!- - -!- - -!

N N N N N N N S N N N N I N N N N N N N I N N N N N N S
/ / / / / W / / / N / / / / / N / / / / / W
C C C C C C C I C C C C P C C C C C C P C C C C C C I
T U U T U T
C T T C C
H L L A T C H

```

TB00670S

## AN21 9-Bit Parity Generator/Checker With 82S153/153A

*Application Note*

### Application Specific Products

#### INTRODUCTION

This application note presents the design of a parity generator using Signetics PLD, 82S153 or 82S153A, which enables the designers to customize their circuits in the form of "sum-of-products". The PLA architecture and the 10 bi-directional I/O's make it possible to implement the 9-bit parity generator/checker in one chip without any external wiring between pins. A logic diagram of the device is shown in Appendix A.

The parity of an 8-bit word is generated by counting the number of "1's" in the word. If the number is odd, the word has odd parity. If the number is even, the word has even parity. Thus, a parity generator designed for even parity, for example, will generate a "0" if the parity is even, or a "1" if parity is odd. Conversely, an odd parity generator will generate a "0" if the parity of the word is odd, or a "1" if the parity is even. This bit is then concatenated to the word making it 9-bits

long. When the word is used elsewhere, its parity may be checked for correctness.

#### FEATURES

- Generates even and odd parities ( $SUM_E$  and  $SUM_O$ )
- $SUM_E = "1"$  for even parity, "0" for odd parity
- $SUM_O = "1"$  for odd parity, "0" for even parity
- Generate parity or check for parity errors
- Cascaded to expand word length

#### DESCRIPTION

The most straight forward way of implementing the parity generator/checker is to take the 9-input truth table (8 inputs for the 8-bit word, and 1 input for cascading the previous stage) and put it in a  $256 \times 4$  PROM. Since there are  $2^9$  combinations and half of them is odd,

the other half is even, the circuit will take 256 terms. An alternative is to divide the 9-bits into 3 groups of 3-bits as shown in Figure 1. If the sum of the 3-bits is odd, then the intermediate output  $SU_1$ , or  $SU_2$ , or  $SU_3$  equals 1. Otherwise it equals 0. The intermediate results are grouped together and  $SUM_O$  becomes "1" if the sum is odd, otherwise  $SUM_O$  equals "0". The circuit is implemented using AMAZE as shown in Figure 3.  $SU_1$  is an intermediate output for inputs  $I_0, I_1$  and  $I_2$ . In the same manner,  $SU_2$  and  $SU_3$  are intermediate outputs for  $I_3, I_4, I_5$  and  $I_6, I_7, I_8$ . The design uses up 16 product terms and 5 control terms leaving 16 product terms and 4 bi-directional I/O's to implement other logic designs.

The design is tested by using the logic simulator provided by AMAZE. The input test vector is chosen to exhaustively test for all 8 input combinations at all 4 sections of the circuit.

# 9-Bit Parity Generator/Checker With 82S153/153A

AN21

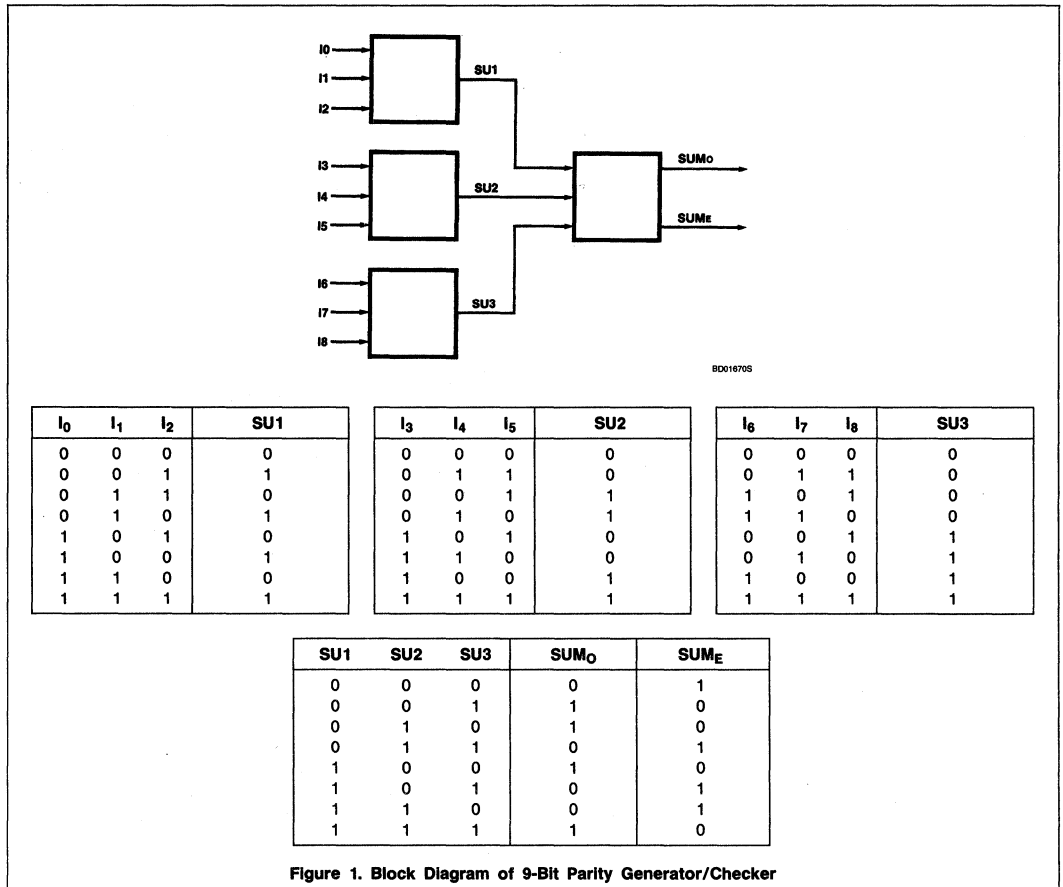


Figure 1. Block Diagram of 9-Bit Parity Generator/Checker

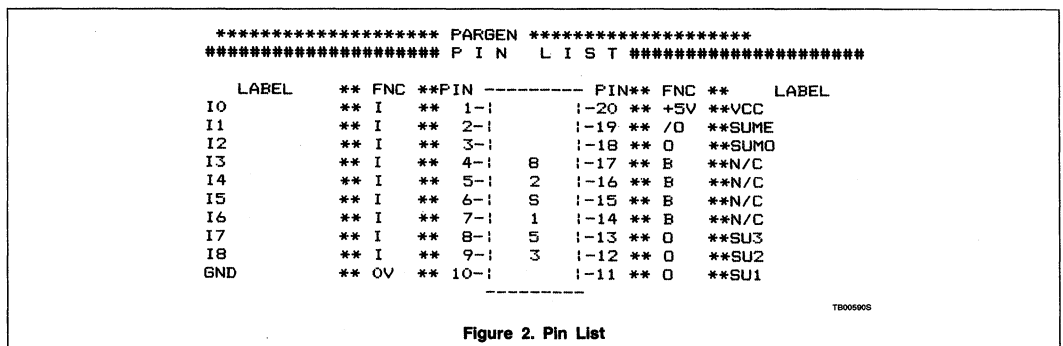


Figure 2. Pin List

# 9-Bit Parity Generator/Checker With 82S153/153A

AN21

```

***** PARGEN *****
@DEVICE TYPE
82S153
@DRAWING
***** PARITY GENERATOR/CHECKER
@REVISION
***** REV. -
@DATE
***** xx/xx/xxxx
@SYMBOL
***** FILE ID: PARGEN
@COMPANY
***** SIGNETICS
@NAME
@DESCRIPTION
*****
* This circuit is a 9-bit parity generator/checker commonly used *
* for error detection in high speed data transmission/retrieval. *
* The odd parity output (SUMO) is high when the sum of the data *
* bits is odd. Otherwise it is low. *
* The even parity output (SUME) is high when the sum of the data *
* bits is even. It is low otherwise. *
*****
@COMMON PRODUCT TERM
@I/O DIRECTION
"
*****
* SU1, SU2 and SU3 are outputs which are defined in the PIN LIST *
* and therefore they don't need to be defined here again. *
*****
@OUTPUT POLARITY
"
*****
* The output polarities of different outputs are defined in the *
* PIN LIST. They don't have to be defined again here. *
*****
@LOGIC EQUATION
"
*****
* SU1, SU2, and SU3 are intermediate terms *
*****

```

INPUTS			OUTPUTS	
SU3	SU2	SU1	SUMO	SUME = /SUMO
18	17	16	SU3	
15	14	13	SU2	
12	11	10	SU1	
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	0

```

"
SU1 = /12 * /11 * /10 + /12 * /11 * /10 +
12 * /11 * /10 + /12 * /11 * /10 ;
SU2 = /15 * /14 * /13 + /15 * /14 * /13 +
15 * /14 * /13 + /15 * /14 * /13 ;
SU3 = /18 * /17 * /16 + /18 * /17 * /16 +
18 * /17 * /16 + /18 * /17 * /16 ;
SUMO = /SU1 * /SU2 * /SU3 + /SU1 * /SU2 * /SU3 +
SU1 * /SU2 * /SU3 + SU1 * /SU2 * /SU3 ;
SUME = /( /SU1 * /SU2 * /SU3 + /SU1 * /SU2 * /SU3 +
SU1 * /SU2 * /SU3 + SU1 * /SU2 * /SU3 ) ;

```

T8009805

Figure 3. AMAZE Implementation of the Parity Generator/Checker Circuit

# 9-Bit Parity Generator/Checker With 82S153/153A

AN21

Table 1. Programming Table

```

***** FARGEN *****
Cust/Project -
Date - ***** XX/XX/XXXX
Rev/I. D. - ***** REV. -

82S153 ! POLARITY !
-----!-----!-----!
T ! !L:H:H:H:H:H:H:H:H:H!
E !-----!-----!-----!
R ! I ! B(i) ! B(o) !
M !-----!-----!-----!
  !7_6_5_4_3_2_1_0!9_8_7_6_5_4_3_2_1_0!9_8_7_6_5_4_3_2_1_0!
0! - - - - , L L H! - - - - - - - - - !. . A A A A . . A A !
1! - - - - , L H L! - - - - - - - - - !. . A A A A . . A A !
2! - - - - , H L L! - - - - - - - - - !. . A A A A . . A A !
3! - - - - , H H H! - - - - - - - - - !. . A A A A . . A A !
4! - L L , H - - - - - - - - - - - !. . A A A A . . A A !
5! - L H , L - - - - - - - - - - - !. . A A A A . . A A !
6! - H L , L - - - - - - - - - - - !. . A A A A . . A A !
7! - H H , H - - - - - - - - - - - !. . A A A A . . A A !
8! L H - - - - - - - - - - - - - - - !. . A A A A . . A A !
9! H L - - - - - - - - - - - - - - - !. . A A A A . . A A !
10! L L - - - - - - - - - - - - - - - !. . A A A A . . A A !
11! H H - - - - - - - - - - - - - - - !. . A A A A . . A A !
12! - - - - , - - - - - - - - - - - , H L L - ! A A A A A A . . A A !
13! - - - - , - - - - - - - - - - - , L H L - ! A A A A A A . . A A !
14! - - - - , - - - - - - - - - - - , L H L - ! A A A A A A . . A A !
15! - - - - , - - - - - - - - - - - , H H H - ! A A A A A A . . A A !
16! 0 0 0 0, 0 0 0 0! 0 0 0 0 0 0 0 0 0 0! 0! A A A A A A A A A A !
17! 0 0 0 0, 0 0 0 0! 0 0 0 0 0 0 0 0 0 0! 0! A A A A A A A A A A !
18! 0 0 0 0, 0 0 0 0! 0 0 0 0 0 0 0 0 0 0! 0! A A A A A A A A A A !
19! 0 0 0 0, 0 0 0 0! 0 0 0 0 0 0 0 0 0 0! 0! A A A A A A A A A A !
20! 0 0 0 0, 0 0 0 0! 0 0 0 0 0 0 0 0 0 0! 0! A A A A A A A A A A !
21! 0 0 0 0, 0 0 0 0! 0 0 0 0 0 0 0 0 0 0! 0! A A A A A A A A A A !
22! 0 0 0 0, 0 0 0 0! 0 0 0 0 0 0 0 0 0 0! 0! A A A A A A A A A A !
23! 0 0 0 0, 0 0 0 0! 0 0 0 0 0 0 0 0 0 0! 0! A A A A A A A A A A !
24! 0 0 0 0, 0 0 0 0! 0 0 0 0 0 0 0 0 0 0! 0! A A A A A A A A A A !
25! 0 0 0 0, 0 0 0 0! 0 0 0 0 0 0 0 0 0 0! 0! A A A A A A A A A A !
26! 0 0 0 0, 0 0 0 0! 0 0 0 0 0 0 0 0 0 0! 0! A A A A A A A A A A !
27! 0 0 0 0, 0 0 0 0! 0 0 0 0 0 0 0 0 0 0! 0! A A A A A A A A A A !
28! 0 0 0 0, 0 0 0 0! 0 0 0 0 0 0 0 0 0 0! 0! A A A A A A A A A A !
29! 0 0 0 0, 0 0 0 0! 0 0 0 0 0 0 0 0 0 0! 0! A A A A A A A A A A !
30! 0 0 0 0, 0 0 0 0! 0 0 0 0 0 0 0 0 0 0! 0! A A A A A A A A A A !
31! 0 0 0 0, 0 0 0 0! 0 0 0 0 0 0 0 0 0 0! 0! A A A A A A A A A A !
D9! - - - - - - - - - - - - - - - - - ! . . . . .
D8! - - - - - - - - - - - - - - - - - ! . . . . .
D7! 0 0 0 0, 0 0 0 0! 0 0 0 0 0 0 0 0 0 0! . . . . .
D6! 0 0 0 0, 0 0 0 0! 0 0 0 0 0 0 0 0 0 0! . . . . .
D5! 0 0 0 0, 0 0 0 0! 0 0 0 0 0 0 0 0 0 0! . . . . .
D4! 0 0 0 0, 0 0 0 0! 0 0 0 0 0 0 0 0 0 0! . . . . .
D3! - - - - - - - - - - - - - - - - - ! . . . . .
D2! - - - - - - - - - - - - - - - - - ! . . . . .
D1! - - - - - - - - - - - - - - - - - ! . . . . .
D0! 0 0 0 0, 0 0 0 0! 0 0 0 0 0 0 0 0 0 0! . . . . .

  I I I I I I I S S N N N N S S S I S S N N N S S S I
  7 6 5 4 3 2 1 0 U U / / / U U U B U U / / / U U U B
                M M C C C C 3 2 1 M M C C C C 3 2 1
                E O E O

                                          TB008105
    
```

## 9-Bit Parity Generator/Checker With 82S153/153A

AN21

```

"
*****
* This is a test pattern for the 9-bit parity generator/checker *
* circuit. The simulator will use this file as an input to      *
* simulate the logical function.                                *
*****

"          SS          EXPECTED
          UU          SSS          OUTPUTS
"IIIIIIII MMBBBBUUUI - BBBB
"76543210 E076543218 98321
LLLLLLLL //L
HLHLHL //H "LHLH
LHLHL //H "LHLH
HHLHL //L "HLLH
LLHLHL //H "LHLL
HLLHL //L "HLHL
LHLHL //L "HLHL
HHHHH //H "LHHH
QUIT

```

TB00003

## a. Input Pattern PARGEN.TST

```

82S153 A:pargen.STD
" This file is the result of logic simulation of the parity generator/checker
" circuit. The inputs are read from input file PARGEN.TST
"
" INPUTS <=B(I/O)=> TRACE TERMS
" 76543210 9876543210
"
00000000 HL...LLO ;
10110100 LH...LH1 ;
01100110 LH...LH1 ;
11010010 HL...LH0 ;
00101101 LH...LH1 ;
10011001 HL...LH0 ;
01001011 HL...LH0 ;
11111111 LH...LH1 ;
"
" ----- I/O CONTROL LINES
" 00IIII000I DESIGNATED I/O USAGE
" 00IIII000I ACTUAL I/O USAGE
"
" PIN LIST...
" 08 07 06 05 04 03 02 01 19 18 17 16 15 14 13 12 11 09 ;

```

TB00015

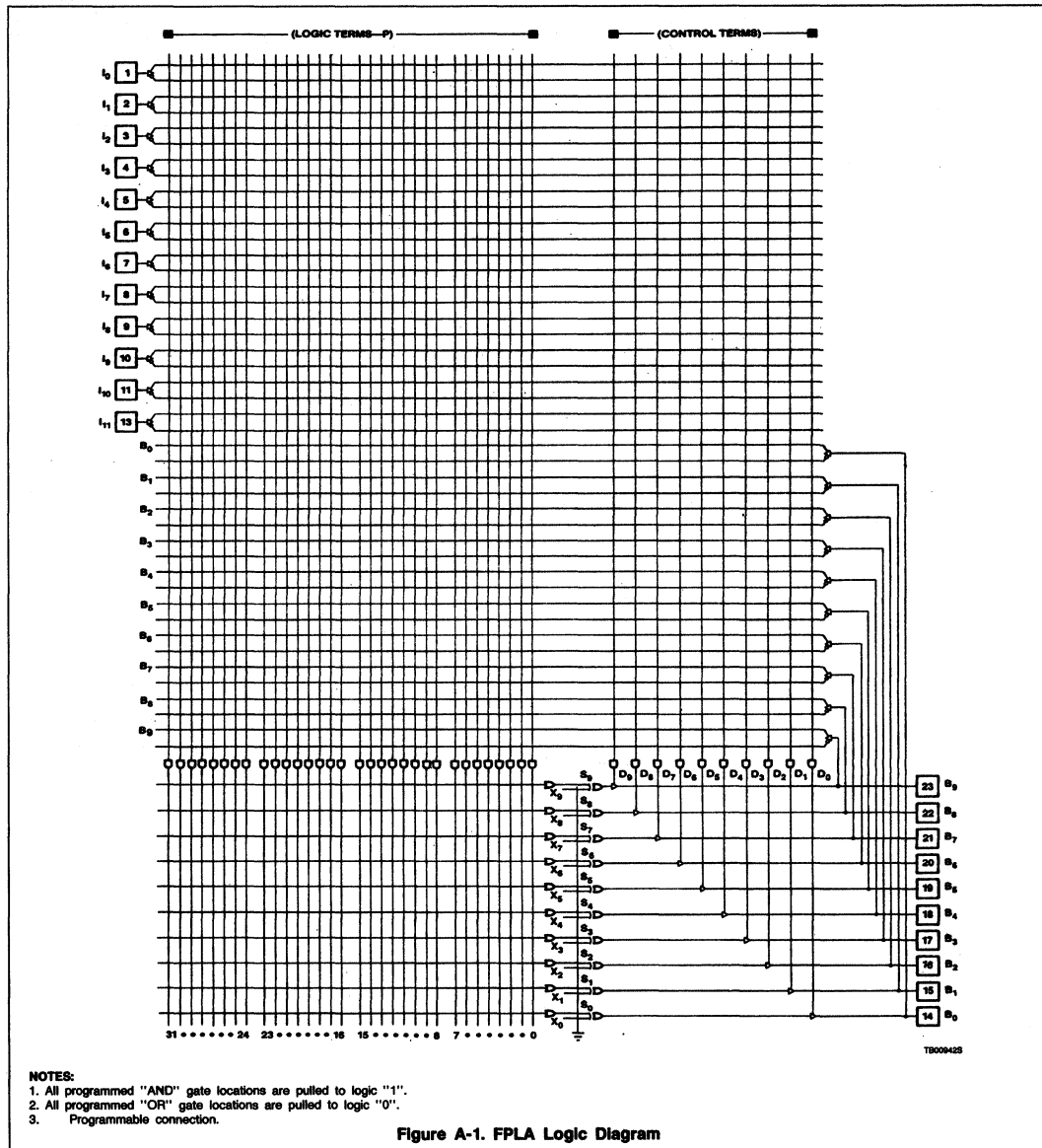
## b. Output File From SIMULATOR

Figure 4. Test Vectors

# 9-Bit Parity Generator/Checker With 82S153/153A

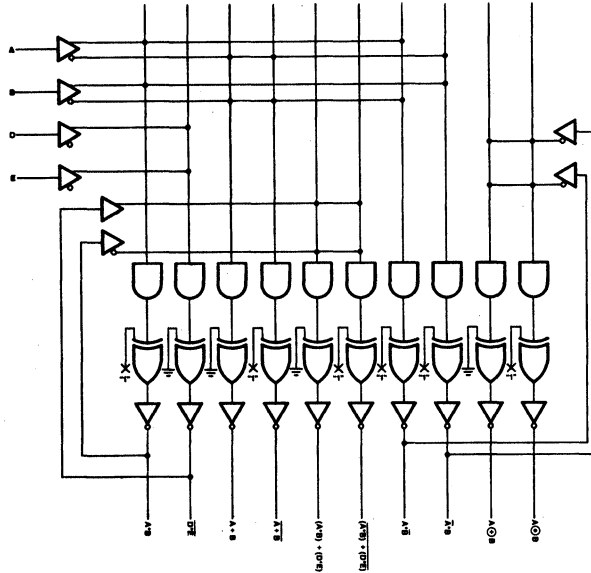
AN21

## APPENDIX A



- NOTES:**
1. All programmed "AND" gate locations are pulled to logic "1".
  2. All programmed "OR" gate locations are pulled to logic "0".
  3. Programmable connection.

Figure A-1. FPLA Logic Diagram



80018805

a. Basic Gates Configuration

```

***** TK151-1 *****
***** P I N   L I S T *****

```

LABEL	** FNC	**PIN	-----	PIN**	FNC **	LABEL
A	** I	** 1-1		1-20	** +5V	**VCC
B	** I	** 2-1		1-19	** 0	**A_NOR_B
D	** I	** 3-1		1-18	** /0	**D_OR_B
E	** I	** 4-1	8	1-17	** /0	**D_NAND_E
N/C	** I	** 5-1	2	1-16	** 0	**A_AND_B
N/C	** I	** 6-1	5	1-15	** /0	**AB_OR_DE
N/C	** /B	** 7-1	1	1-14	** 0	**AB_NOR_DE
N/C	** /B	** 8-1	5	1-13	** /0	**A_XOR_B
A-N_B	** 0	** 9-1	1	1-12	** 0	**A_EQU_B
GND	** 0V	** 10-1		1-11	** 0	**A_N_B-

T8004405

b. Pin List

Figure 1



PLS151 Toolkit

AN22

```

***** TK151-1 *****
@DEVICE TYPE
***** S2S151
@DRAWING
***** FIGURE 1. BASIC GATE CONFIGURATIONS
@REVISION
***** TK151-1 REV. 0
@DATE
***** JULY 10, 1985
@SYMBOL
***** TK151-1
@COMPANY
***** SIGNETICS
@NAME
***** DAVID K. WONG
@DESCRIPTION
This file contains the 6 basic logic gates: AND, NAND, OR, NOR, XOR and EQU.
@COMMON PRODUCT TERM
@I/O DIRECTION
@OUTPUT POLARITY
@I/O STEERING
@LOGIC EQUATION
A_AND_B = A * B ;
D_NAND_E = /( D * E ) ;
A_OR_B = /( /A * /B ) ;
A_NOR_B = /A * /B ;
AB_OR_DE = /( /(A_AND_B) * D_NAND_E ) ;
AB_NOR_DE = /(A_AND_B) * D_NAND_E ;
A_N_B- = A * /B ;
A-N_B = /A * B ;
A_XOR_B = /( (A_N_B-) * /(A-N_B) ) ; " Equivalent to (A * /B) + (/A * B) "
A_EQU_B = /(A_N_B-) * /(A-N_B) ; " Equivalent to (/A * /B) + (A * B) "
    
```

TB00805

c. Boolean Equations of Basic Gate Configurations

Figure 1 (Continued)

Table 1. Program Table of Basic Gate Configurations

```

***** TK151-1 *****
Cust/Project - ***** DAVID K. WONG
Date - ***** JULY 10, 1985
Rev/I. D. - ***** TK151-1 REV. 0

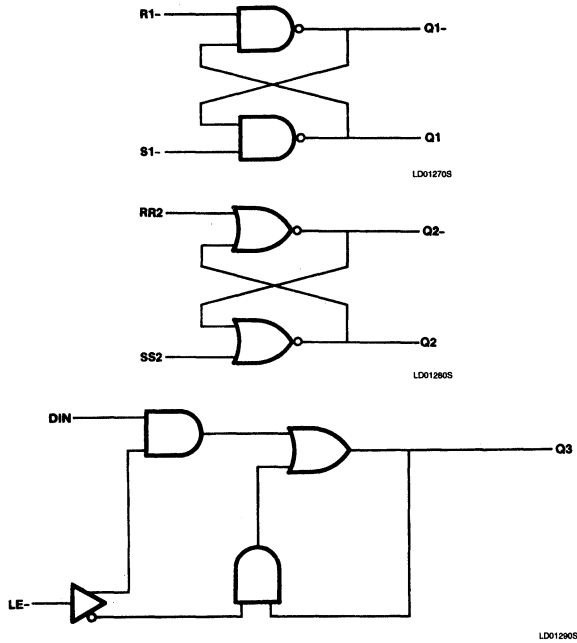
S2S151 DIR !A:A:A:A:A:A:A:A:A:A:A:A:A:A:A:A
-----
T ! POL !H:L:L:L:L:L:L:L:L:H:H:H:L:L!
E !-----
R ! I ! B(i) !
M ! !1 1 -----
15_4_3_2_1_0!1_0_9_8_7_6_5_4_3_2_1_0!
0!0_0_0_0_0_0!0_0_0_0_0_0_0_0_0_0_0_0!
1!0_0_0_0_0_0!0_0_0_0_0_0_0_0_0_0_0_0!
2!-,-,-, H L!-,-,-,-,-,-,-,-,-,-!
3!-,-,-, L H!-,-,-,-,-,-,-,-,-,-!
4!-,-,-,-,-,-,-,-,-,-,-,-,-,-,-!
5!-,-,-,-,-,-,-,-,-,-,-,-,-,-,-!
6!-,-,-,-,-,-,-,-,-,-,-,-,-,-,-!
7!-,-,-,-,-,-,-,-,-,-,-,-,-,-,-!
8!-,-,-,-,-,-,-,-,-,-,-,-,-,-,-!
9!-,-,-,-,-,-,-,-,-,-,-,-,-,-,-!
10!-,-,-,-,-,-,-,-,-,-,-,-,-,-,-!
11!-,-,-,-,-,-,-,-,-,-,-,-,-,-,-!
D2!-,-,-,-,-,-,-,-,-,-,-,-,-,-,-!
D1!-,-,-,-,-,-,-,-,-,-,-,-,-,-,-!
D0!-,-,-,-,-,-,-,-,-,-,-,-,-,-,-!

N N E D B A A A D A A A A A A A N N
/ / - - - B B - - - / /
C C N O N A - - - X E N - - - C C
D R A N D N O R N
R - N D R O R U B -
B B D - - - R B B - B
E E E D B B
E
    
```

TB00805

PLS151 Toolkit

AN22



a. Basic Latches Configuration

```

***** TK151-2 *****
***** P I N   L I S T *****

```

LABEL	** FNC	**PIN	PIN	** FNC	** LABEL
R1-	** I	** 1-	1-20	** +5V	**VCC
S1-	** I	** 2-	1-19	** /0	**Q1
RR2	** I	** 3-	1-18	** /0	**Q1-
SS2	** I	** 4-	8	1-17	** 0 **Q2
DIN	** I	** 5-	2	1-16	** 0 **Q2-
LE-	** I	** 6-	S	1-15	** /0 **Q3
N/C	** /B	** 7-	1	1-14	** 0 **D_N_LE
N/C	** /B	** 8-	5	1-13	** 0 **Q3_N_LE-
N/C	** /B	** 9-	1	1-12	** /B **N/C
GND	** OV	** 10-	1-11	** /B	**N/C

b. Pin List

Figure 2

PLS151 Toolkit

AN22

```

***** TK151-2 *****
@DEVICE TYPE
S2S151
@DRAWING
***** FIGURE 2. BASIC LATCHES CONFIGURATION
@REVISION
***** TK151-2 REV. 0
@DATE
***** JULY 10, 1985
@SYMBOL
***** TK151-2
@COMPANY
***** SIGNETICS
@NAME
***** DAVID K. MONG
@DESCRIPTION
This files contains basic configurations of two RS-latches and a D-latch.
@COMMON PRODUCT TERM
@I/O DIRECTION
@OUTPUT POLARITY
@I/O STEERING
@LOGIC EQUATION

" *****
* CROSS-COUPLED NAND RS-LATCH *
***** "
Q1- = / ( /R1 * Q1 ) ;
Q1 = / ( /S1 * Q1- ) ;

" *****
* CROSS-COUPLED NOR RS-LATCH *
***** "
Q2- = RR2 * Q2 ;
Q2 = SS2 * Q2- ;

" *****
* D-LATCH *
***** "
D_N_LE = DIN * LE- ;
Q3_N_LE = Q3 * /LE- ;
Q3 = / ( /D_N_LE ) * / ( Q3_N_LE- ) ;
    
```

TB00600

c. Boolean Equations of Basic Latches Configuration

Figure 2 (Continued)

Table 2. Program Table of Basic Latches

```

***** TK151-2 *****
Cust/Project - ***** DAVID K. MONG
Date - ***** JULY 10, 1985
Rev/I. D. - ***** TK151-2 REV. 0

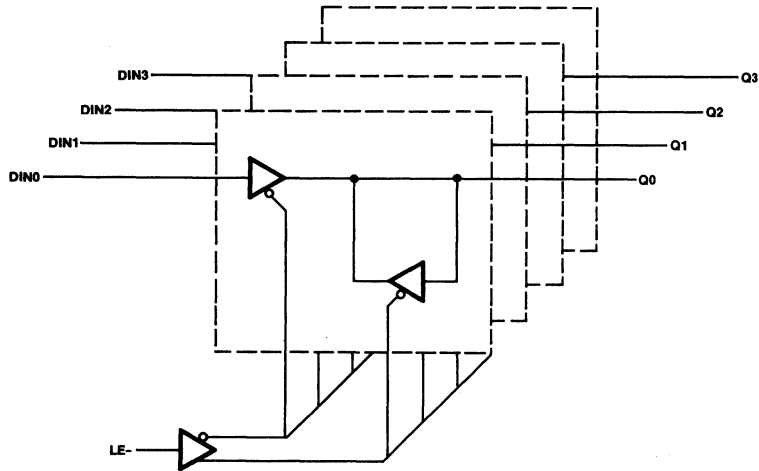
826151 DIR !A:A:A:A:A:A:A:A:A:A!
T ! L L L H H L L H H L L L L L L !
E !
R ! I ! B(i) !
N !-----111-----!
-----5_4_3_2_1_0!1_0_9_8_7_6_5_4_3_2_1_0!
0!0 0,0 0 0 0!0 0 0 0,0 0 0,0 0 0 0!
1!0 0,0 0 0 0!0 0 0 0,0 0 0,0 0 0 0!
2!0 0,0 0 0 0!0 0 0 0,0 0 0,0 0 0 0!
3!0 0,0 0 0 0!0 0 0 0,0 0 0,0 0 0 0!
4!0 0,0 0 0 0!0 0 0 0,0 0 0,0 0 0 0!
5!L - - - - -! - - - - -!H - - - - -!
6!H - - - - -! - - - - -! - - - - -!
7! - - - - -! - - - - -!L L - - - - -!
8! - - - - -!H - - - - -!H - - - - -!
9! - - - - -!H - - - - -!H - - - - -!
10! - - - - -!L!H - - - - -! - - - - -!
11! - - - - -!L - - - - -!H - - - - -!
D2!0 0,0 0 0 0!0 0 0 0,0 0 0,0 0 0 0!
D1! - - - - -! - - - - -! - - - - -!
DO! - - - - -! - - - - -! - - - - -!

L D S R S R Q Q Q Q Q D Q N N N N N
E I B R 1 1 1 1 2 2 3 _ 3 / / / / /
- N 2 2 - - - - - N _ C C C C C
_ N
L _
E L
E
    
```

TB007108

PLS151 Toolkit

AN22



LD013005

a. 4-Bit D-Latch

\*\*\*\*\* TK151-3 \*\*\*\*\*  
 \*\*\*\*\* P I N L I S T \*\*\*\*\*

LABEL	** FNC	**PIN	PIN	** FNC	** LABEL
DIN0	** I	** 1-!	1-20	** +SV	**VCC
DIN1	** I	** 2-!	1-19	** B	**Q0
DIN2	** I	** 3-!	1-18	** B	**Q1
DIN3	** I	** 4-!	8	1-17	** B **Q2
LE-	** I	** 5-!	2	1-16	** B **Q3
N/C	** I	** 6-!	S	1-15	** B **FB0
N/C	** /B	** 7-!	1	1-14	** B **FB1
N/C	** /B	** 8-!	5	1-13	** B **FB2
N/C	** /B	** 9-!	1	1-12	** B **FB3
GND	** 0V	** 10-!	1-11	** /B	**N/C

TB007005

b. Pin List

Figure 3

PLS151 Toolkit

AN22

```

***** TK151-3 *****
@DEVICE TYPE
82S151
@DRAWING
***** FIGURE 3. 4 BIT D-LATCH
@REVISION
***** TK151-3 REV. 0
@DATE
***** JULY 10, 1985
@SYMBOL
***** TK151-3
@COMPANY
***** SIGNETICS
@NAME
***** DAVID K. WONG
@DESCRIPTION
This file contains a 4-bit D-latch. External jumper is required for each latch
between Qn and FBn. When LE- is HIGH, outputs Q0, Q1, Q2, and Q3 are active and
outputs FB0, FB1, FB2, and FB3 are tri-stated. Qn, therefore, equals Dn when
LE- is HIGH. When LE- becomes LOW, Q outputs are tri-stated, and FB outputs are
active. Since FB outputs feedback to their own inputs, they are latched to
whatever level the Q outputs were in when LE- goes from HIGH to LOW.
@COMMON PRODUCT TERM
@I/O DIRECTION
D0 = /( /LE- ) ;
D1 = /( /LE- ) ;
@OUTPUT POLARITY
@I/O STEERING
@LOGIC EQUATION
Q0 = DINO ;
Q1 = DIN1 ;
Q2 = DIN2 ;
Q3 = DIN3 ;
FB0 = Q0 ;
FB1 = Q1 ;
FB2 = Q2 ;
FB3 = Q3 ;
    
```

TB00690S

c. Boolean Equations of 4-Bit D-Latch

Figure 3 (Continued)

Table 3. Program Table of 4-Bit D-Latch

```

***** TK151-3 *****
Cust/Project - ***** DAVID K. WONG
Date - ***** JULY 10, 1985
Rev/I. D. - ***** TK151-3 REV. 0

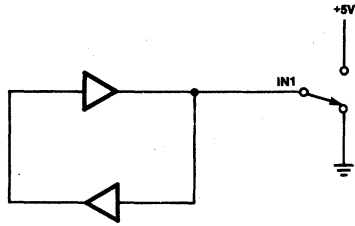
82S151 DIR !A:A:A:A:A:A:A:A:A:A!
T ! PDL !H:H:H:H:H:H:H:H:L:L:L!
E !
R ! I ! B(i) !
M !:1 1 !
:5 4 3 2 1 0:1 0 9 8 7 6 5 4 3 2 1 0:
0:0 0 0 0 0 0:0 0 0 0 0 0 0 0 0 0 0:
1:0 0 0 0 0 0:0 0 0 0 0 0 0 0 0 0 0:
2:0 0 0 0 0 0:0 0 0 0 0 0 0 0 0 0 0:
3:0 0 0 0 0 0:0 0 0 0 0 0 0 0 0 0 0:
4:- - - - -! - - H - - - - -!
5:- - - - -! - - H - - - - -!
6:- - - - -! - - H - - - - -!
7:- - - - -! - - H - - - - -!
8:- - H - -! - - - - -!
9:- - H - -! - - - - -!
10:- - H - -! - - - - -!
11:- - - - -! - - H - - - - -!
B2:0 0 0 0 0 0:0 0 0 0 0 0 0 0 0 0 0:
D1: H - - - -! - - - - -!
DO:- L - - - -! - - - - -!

N L D D D D Q Q Q Q F F F F N N N N
/ E I I I I 0 1 2 3 B B B B / / / /
C - N N N N 0 1 2 3 C C C C
3 2 1 0
    
```

TB00740S

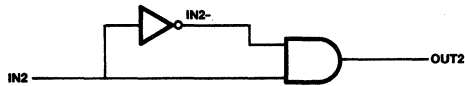
PLS151 Toolkit

AN22



TC016005

Switch Debouncer



LD013105

Pulse Shaper

a. Single Pin Switch Debouncer and Pulse Shaper

\*\*\*\*\* TK151-4 \*\*\*\*\*  
 \*\*\*\*\* P I N L I S T \*\*\*\*\*

LABEL	** FNC	**PIN	-----	PIN**	FNC **	LABEL
IN2	** I	** 1-1		1-20	** +5V	**VCC
N/C	** I	** 2-1		1-19	** 0	**IN1
N/C	** I	** 3-1		1-18	** /0	**IN2-
N/C	** I	** 4-1	8	1-17	** 0	**OUT2
N/C	** I	** 5-1	2	1-16	** /B	**N/C
N/C	** I	** 6-1	6	1-15	** /B	**N/C
N/C	** /B	** 7-1	1	1-14	** /B	**N/C
N/C	** /B	** 8-1	5	1-13	** /B	**N/C
N/C	** /B	** 9-1	1	1-12	** /B	**N/C
GND	** OV	** 10-1		1-11	** /B	**N/C

T6007305

b. Pin List

Figure 4

PLS151 Toolkit

AN22

```

***** TK151-4 *****
@DEVICE TYPE
@S2S151
@DRAWING
***** PLS151 TOOLKIT FIGURE 4
@REVISION
***** tk151-4 REV. 0
@DATE
***** JULY 10, 1985
@SYMBOL
***** TK151-4
@COMPANY
***** SIGNETICS
@NAME
***** DAVID K. WONG
@DESCRIPTION
This file contains a single pin debouncer and a pulse shaper.
@COMMON PRODUCT TERM
@I/O DIRECTION
@OUTPUT POLARITY
@I/O STEERING
@LOGIC EQUATION
" *****
 * DEBOUNCER -- This circuit uses one bidirectional pin to de- *
 * bounce a single pole double throw switch. When *
 * the switch is thrown to its opposit polarity, *
 * it shorts the output to Vcc or ground. One pro- *
 * pagation delay later, the output is switched to *
 * the same polarity as the switch. *
 ***** "
INI = IN1 ;

" *****
 * PULSE SHAPER -- This circuit creates a LOW-to-HGH glitch *
 * which has the pulse width of one prop delay. *
 * Wider pulse width may be generated with add- *
 * itional feedback loops. *
 ***** "
IN2- = /( IN2 ) ;
OUT2 = IN2- * IN2 ;
    
```

T8007205

c. Boolean Equations of Single Pin and Pulse Shaper  
Figure 4 (Continued)

Table 4. Program Table of Single Pin Debouncer and Shaper

```

***** TK151-4 *****
Cust/Project - ***** DAVID K. WONG
Date        - ***** JULY 10, 1985
Rev/I. D.   - ***** tk151-4 REV. 0

S2S151 DIR :A1:A2:A3:A4:A5:A6:A7:A8:A9:A10:
T : PDL :M1:L1:M2:L2:M3:L3:M4:L4:M5:L5:M6:L6:
E :----- I' ! B(i) !
M :----- !1 1
--:5.4.3.2.1.0!1.0.9.8.7.6.5.4.3.2.1.0!
0:0 0,0 0 0 0!0 0 0 0,0 0 0 0,0 0 0 0!
1:0 0,0 0 0 0!0 0 0 0,0 0 0 0,0 0 0 0!
2:0 0,0 0 0 0!0 0 0 0,0 0 0 0,0 0 0 0!
3:0 0,0 0 0 0!0 0 0 0,0 0 0 0,0 0 0 0!
4:0 0,0 0 0 0!0 0 0 0,0 0 0 0,0 0 0 0!
5:0 0,0 0 0 0!0 0 0 0,0 0 0 0,0 0 0 0!
6:0 0,0 0 0 0!0 0 0 0,0 0 0 0,0 0 0 0!
7:0 0,0 0 0 0!0 0 0 0,0 0 0 0,0 0 0 0!
8:0 0,0 0 0 0!0 0 0 0,0 0 0 0,0 0 0 0!
9: - - - - -!H- H - - - - -!
10:- - - - -!H- - - - -!
11:- - - - -!H - - - - -!
D2:0 0,0 0 0 0!0 0 0 0,0 0 0 0,0 0 0 0!
D1:0 0,0 0 0 0!0 0 0 0,0 0 0 0,0 0 0 0!
D0:- - - - -!- - - - -!

N N N N N I I I O N N N N N N N N
/ / / / / N N N U / / / / / / / /
C C C C C 2 1 2 T C C C C C C C C
- 2
    
```

T8006505

### Application Specific Products

#### INTRODUCTION

THE PLS168/168A is a bipolar Field-Programmable Logic Sequencer as shown in Figure 1, which consists of 12 inputs, a 48 product term PLA and 14 R/S flip-flops. Out of the 14 flip-flops, six are buried State Registers ( $P_4 - P_9$ ), four Output Registers ( $F_0 - F_3$ ), and four Dual-purpose Registers ( $P_0 - P_3$ ), which may be used as Output or State Registers. All flip-flops are positive edge-triggered. They are preset to "1" at power-up, or may be asynchronously set to "1" by an optional PR/OE pin, which may be programmed either as a preset pin or as an Output Enable pin. Additional features includes the Complement Array and diagnostics features.

#### ARCHITECTURE

As shown in Figure 2, the device is organized as a decoding AND-OR network which drives a set of registers some of which, in turn, feedbacks to the AND/OR decoder while the rest serve as outputs. Outputs  $P_0$  to  $P_3$  may be programmed to feedback to the AND/OR decoder as State Registers and, at the same time, used as outputs. The user now can design a 10-bit state machine without external wiring. The AND/OR array is the classical PLA structure in which the outputs of all the AND gates can be programmed to drive all

the OR gates. The schematic diagram of the AND-OR array is shown in Figure 3. This structure provides the user a very structured design methodology which can be automated by CAD tools, such as Signetics AMAZE software package. The output of the PLA is in the form of sum-of-products which, together with the RS flip-flops, is the ideal structure for implementation of state machines. (Refer to Appendix A for a brief description of synchronous finite state machines.)

#### Design Tools

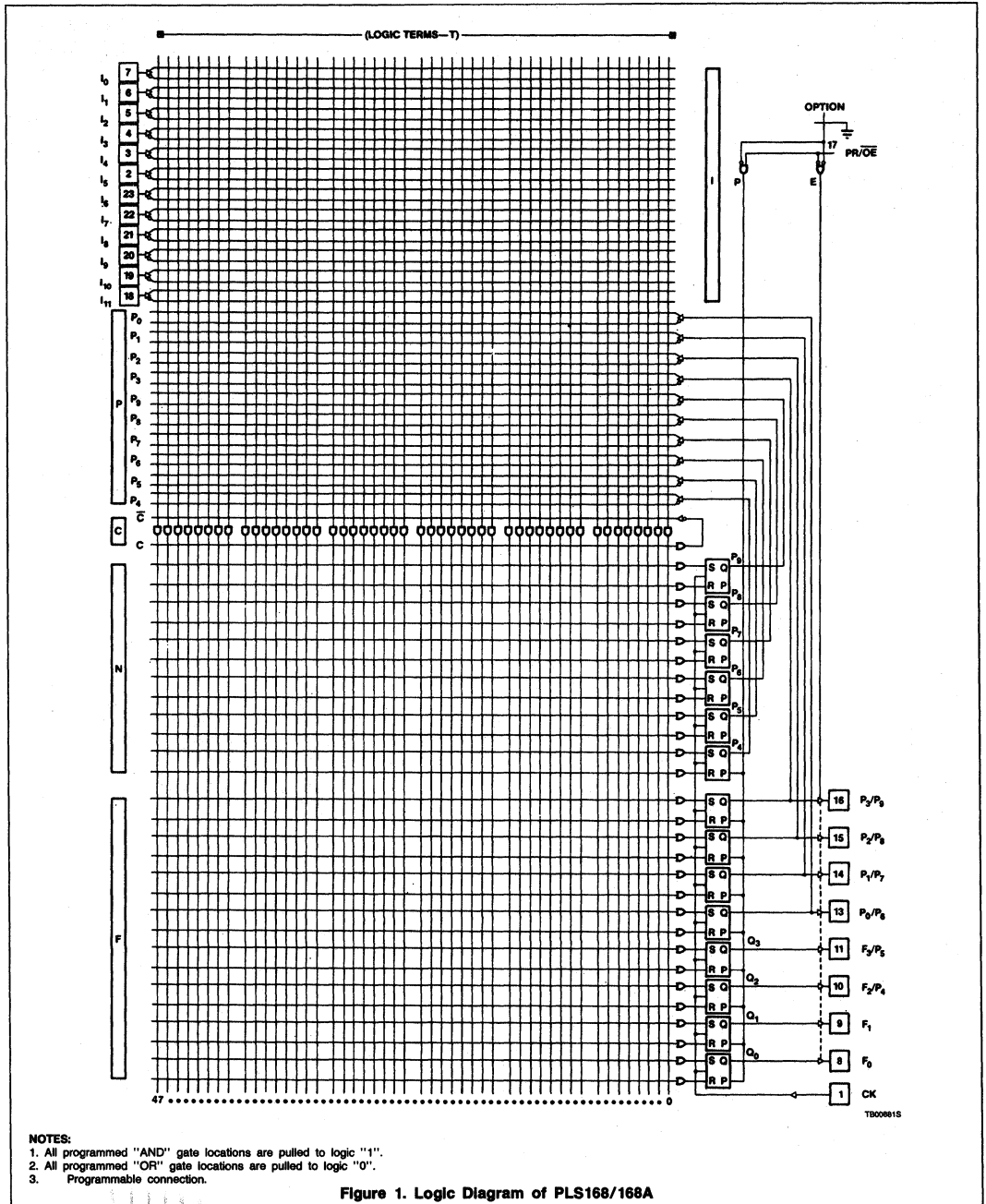
A direct approach to implement a design using the PLS168/168A is the H/L table supplied in the data sheet as shown in Table 1. The table is organized according to input and output of the PLA decoding network. The lefthand side of the table represent the inputs to the AND-array, which includes input from input pins and present state information from the feedback buffers which feedback the contents of the State Register. The righthand side of the table represents the output of the OR-array, which drives the State and Output Registers as the next state and output. Each column in the lefthand side of the table represents an input buffer, which may be inverting, non-inverting, disconnected or unprogrammed. Each column in the righthand side of the table represents a pair of outputs to the flip-flops, which may be set, reset,

disconnected, or unprogrammed. The programming symbols are H, L, -, and 0. (See Figure 4 for details.) For inputs buffers, "H" means that the non-inverting buffer is connected, "L" means that the inverting buffer is connected, "-" means that both inverting and non-inverting buffers are disconnected, and "0" means that both inverting and non-inverting buffers are connected which causes that particular AND-term to be unconditionally Low. On the output side of the table, "H" means that the particular AND-term is connected to the OR-term on the "S" input of the particular flip-flop, "L" means that the AND-term is connected to the "R" side, "-" means that the AND-term is not connected to the flip-flop at all, and "0" means that the AND-term is connected to both the "S" and "R" sides. More details of the symbols and their meanings are shown in Appendix B. Each row in the table represents an AND-term. There are 48 AND-terms in the device. Therefore, there are 48 rows in the table. An example of implementing a transition from one state to another is shown in Figure 4a. The state diagram can be implemented by the PLS168 as shown in Figure 4b. The state diagram is translated into H/L format as shown in Figure 4c. The first column on the lefthand side of the table is for the Complement Array which will be discussed in detail in the next section.



PLS168/168A Primer

AN23



### PLS168/168A Primer

### AN23

Table 1. PLS168/168A Programming Table

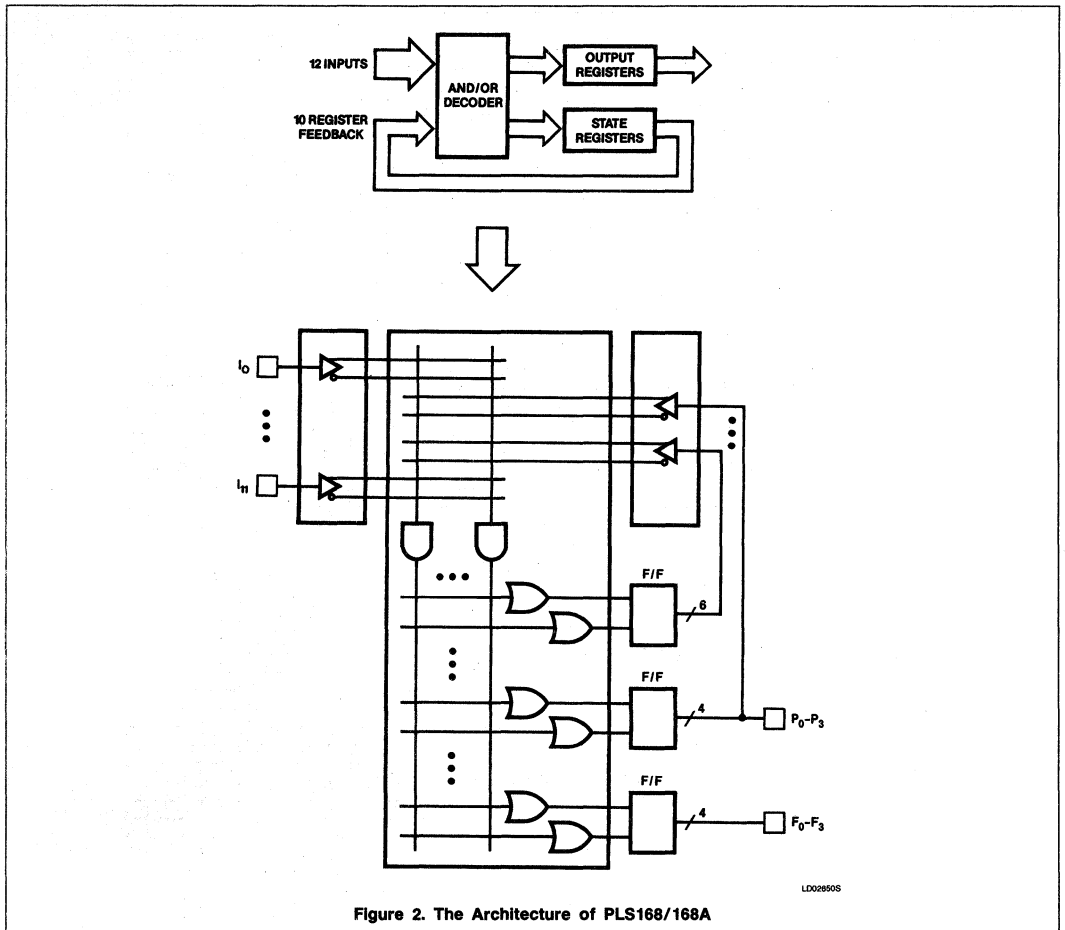
CUSTOMER NAME _____				AND			OR																																		
PURCHASE ORDER # _____				INACTIVE <input type="checkbox"/>			INACTIVE <input type="checkbox"/>																																		
SIGNETICS DEVICE # CF (XXXX)				GENERATE A			SET H																																		
CUSTOMER SYMBOLIZED PART # _____				PROGRAMME +			RESET L																																		
TOTAL NUMBER OF PARTS _____				TRANSPARENT -			NO CHANGE -																																		
PROGRAM TABLE # _____				INACTIVE <input type="checkbox"/>			OPTION																																		
REV _____				L.S. H			PRESET H																																		
DATE _____				L.S. L			OE L																																		
				DONT CARE -			PC																																		
AND							OPTION (PE)																																		
T E R M	C <sub>n</sub>	INPUT (I <sub>m</sub> )										PRESENT STATE (P <sub>n</sub> )				REMARKS	NEXT STATE (N <sub>n</sub> )								OUTPUT (F <sub>i</sub> )																
		1	0	9	8	7	6	5	4	3	2	1	0	9	8		7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3
0																																									
1																																									
2																																									
3																																									
4																																									
5																																									
6																																									
7																																									
8																																									
9																																									
10																																									
11																																									
12																																									
13																																									
14																																									
15																																									
16																																									
17																																									
18																																									
19																																									
20																																									
21																																									
22																																									
23																																									
24																																									
25																																									
26																																									
27																																									
28																																									
29																																									
30																																									
31																																									
32																																									
33																																									
34																																									
35																																									
36																																									
37																																									
38																																									
39																																									
40																																									
41																																									
42																																									
43																																									
44																																									
45																																									
46																																									
47																																									

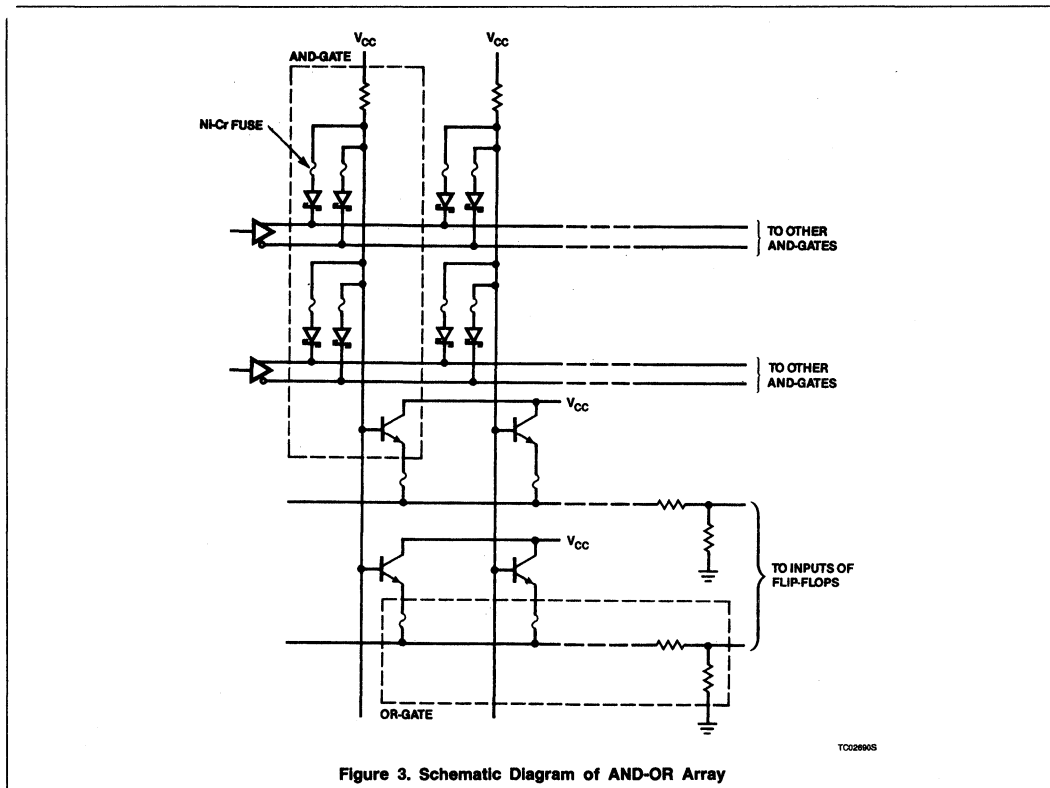
T802005



# PLS168/168A Primer

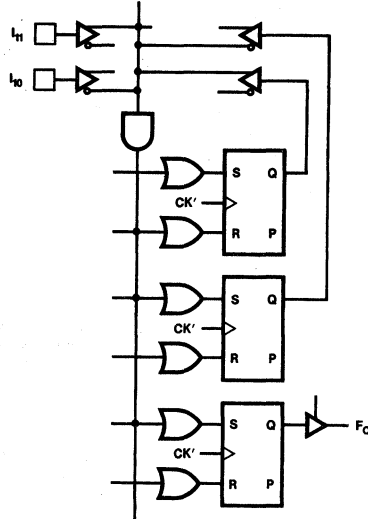
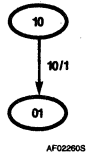
AN23





PLS168/168A Primer

AN23



a. State Diagram

b. Implementation of State Diagram (a) With PLS168

TERM	AND																OPTION (PR/OE)																						
	C	INPUT								PRESENT STATE								OR																					
		1	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	3	2	1	0	
00	-	H	L	-	-	-	-	-	-	-	-	-	-	-	-	-	-	H	L	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	H	-	-	
01																																							
PIN NO.	1	1	8	9	2	2	2	2	2	3	4	5	6	7																									
NAME	IN1	IN0																																				OUT0	

TB0210S

c. PLS168 Programming Table

Figure 4. Implementing State Machine With PLS168

## PLS168/168A Primer

## AN23

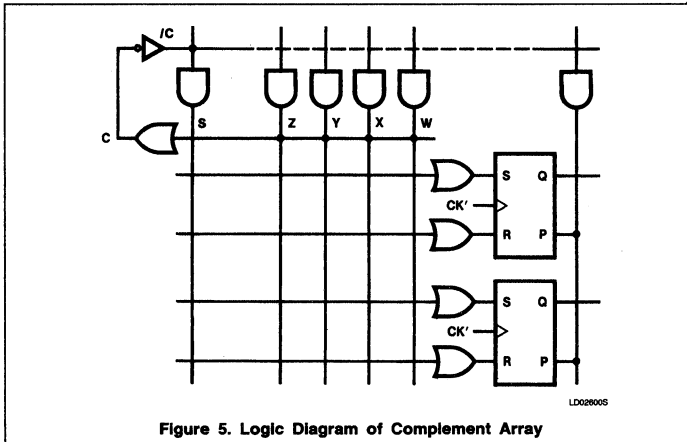


Figure 5. Logic Diagram of Complement Array

**Complement Array**

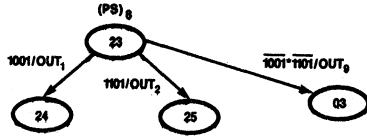
An additional feature is the Complement Array, which is often used to provide escape vectors in case the state machines get into undefined states during power-up or a timing violation due to asynchronous inputs. A logic diagram of the Complement Array is shown in Figure 5. The output of the Complement Array is normally Low when one or more AND-terms are High. If all of the AND-terms are Low, then the output of the Complement

Array will be High. In this example, if each AND-term is a decoder for a particular state and input combination, and if the circuit gets into an undefined state, none of the AND-terms will be High. Therefore, the output /C will be High, which will then enable the AND-term S which in turn may be used to reset all registers to Low or High as predefined. The state machine thus escapes from being in an undefined state by using the Complement Array and one AND-term. Without the Complement Array an alternate way of escaping

from being in an undefined state is by defining all possible states which are not being defined. This method may require quite a few AND-terms depending on the design. Another application for the Complement Array is illustrated by the following example. As shown in Figure 6, when the machine is in state 23, if input vector equals 1001, it will go to next state 24. If the input is 1101, then go to state 25. But if the input is neither 1001 nor 1101, then go to state 03. It takes only two terms to implement the first two transition vectors. To implement the third vector "go to state 03 if input is neither 1001 nor 1101", the Complement Array accepts the outputs of the first two AND-terms as inputs. If the input vector is neither 1001 nor 1101, then both terms will be Low, which causes the output of the Complement Array (/C) to be High. A third AND-term is used to AND state 24 and /C together to set the registers to state 03. The State Diagram is translated into AMAZE syntax as shown in Figure 6b, where all vectors are in square brackets and the Complement Array is represented by the ELSE statement. The State diagram Figure 6a can also be expressed in the format of a program table as shown in Figure 6c. The complement array may be used to exit from different present states to different next states. It can be used many times in one state machine design as shown in Figures 7a, b, and c where the state diagram is implemented using the AMAZE state equation syntax and the H/L format.

PLS168/168A Primer

AN23



NOTE:  $1001 \cdot 1101 = 1001 + 1101$

AF022415

a. State Diagram

WHILE [23]  
 IF [1001] THEN [24] WITH  
 [OUT1]  
 IF [1101] THEN [25] WITH  
 [OUT2]  
 ELSE: [03] WITH [OUT3]

b. AMAZE State Equation Syntax

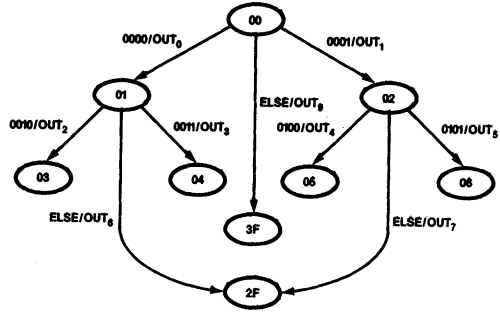
TERM	AND														OPTION (PR/OE)																									
	C	INPUT							PRESENT STATE							NEXT STATE							OUTPUT																	
		1	1	0	0	0	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	3	2	1	0		
00	A	H	L	L	H	-	-	-	-	-	-	-	-	H	L	L	L	H	H	-	-	-	-	H	L	L	H	L	L	-	-	-	-	L	L	L	H			
01	A	H	H	L	H	-	-	-	-	-	-	-	-	H	L	L	L	H	H	-	-	-	-	H	L	L	H	L	H	-	-	-	-	L	L	H	L			
02	?	-	-	-	-	-	-	-	-	-	-	-	-	H	L	L	L	H	H	-	-	-	-	L	L	L	L	H	H	-	-	-	-	H	L	L	H			
03																																								
04																																								
05																																								
PIN NO.	1	1	1	1	1	1	2	2	2	3	4	5	6	7																										
NAME	#N3	#N2	#N1	#N0																																	OP3	OP2	OP1	OP0

c. H/L Format

Figure 6. Application of Complement Array

PLS168/168A Primer

AN23



AF025505

a. State Diagram

```

  WHILE {00}
  IF {0000} THEN {01} WITH {OUT0}
  IF {0001} THEN {02} WITH {OUT1}
  ELSE: {3F} WITH {OUT9}
  WHILE {01}
  IF {0010} THEN {03} WITH {OUT2}
  IF {0011} THEN {04} WITH {OUT3}
  ELSE: {2F} WITH {OUT6}
  WHILE {02}
  IF {0100} THEN {06} WITH {OUT4}
  IF {0101} THEN {06} WITH {OUT5}
  ELSE: {2F} WITH {OUT7}
  
```

b. AMAZE State Equation

T E R M	C	AND														OPTION (PR/OE)																																				
		INPUT								PRESENT STATE						NEXT STATE							OR																													
		1	1	1	1	1	1	1	1	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	3	2	1	0	3	2	1	0															
00	A	L	L	L	L	-	-	-	-	-	-	-	-	-	-	-	-	-	L	L	L	L	L	L	-	-	-	-	-	-	-	-	L	L	L	L	L	L	L	H	-	-	-	-	-	-	-	-	-	-	-	
01	A	L	L	L	L	H	-	-	-	-	-	-	-	-	-	-	-	-	L	L	L	L	L	L	L	-	-	-	-	-	-	-	-	L	L	L	L	H	L	-	-	-	-	-	-	-	-	-	-	-	-	-
02	*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	L	L	L	L	L	L	L	-	-	-	-	-	-	-	-	H	H	H	H	H	H	H	-	-	-	-	-	-	-	-	-	-	-	-
03																																																				
04	A	L	L	H	L	-	-	-	-	-	-	-	-	-	-	-	-	-	L	L	L	L	H	L	-	-	-	-	-	-	-	-	L	L	L	L	H	H	-	-	-	-	-	-	-	-	-	-	-	-	-	
05	A	L	L	H	H	-	-	-	-	-	-	-	-	-	-	-	-	-	L	L	L	L	H	L	-	-	-	-	-	-	-	-	L	L	L	H	L	L	-	-	-	-	-	-	-	-	-	-	-	-	-	
06	*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	L	L	L	L	L	H	-	-	-	-	-	-	-	-	H	L	H	H	H	H	-	-	-	-	-	-	-	-	-	-	-	-	-	
07																																																				
08	A	L	H	L	L	-	-	-	-	-	-	-	-	-	-	-	-	-	L	L	L	L	H	L	-	-	-	-	-	-	-	-	L	L	L	H	L	H	-	-	-	-	-	-	-	-	-	-	-	-	-	
09	A	L	H	L	H	-	-	-	-	-	-	-	-	-	-	-	-	-	L	L	L	L	H	L	-	-	-	-	-	-	-	-	L	L	L	H	H	L	-	-	-	-	-	-	-	-	-	-	-	-	-	
10	*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	L	L	L	L	H	L	-	-	-	-	-	-	-	-	H	L	H	H	H	H	-	-	-	-	-	-	-	-	-	-	-	-	-	
11																																																				
PIN NO.		1	1	1	2	2	2	2	2	2	3	4	5	6	7			1	1	1	1	1	1	1	1	1	1	1	1	8	9	0	1	2	3	4																
NAME		IN8	IN9	IN0	IN1	IN2	IN3	IN4	IN5	IN6	IN7						OP6	OP5	OP4	OP3	OP2	OP1	OP0																													

TB019905

c. PLS168 Programming Table  
 Figure 7. Applications of Complement Array



PLS168/168A Primer

AN23

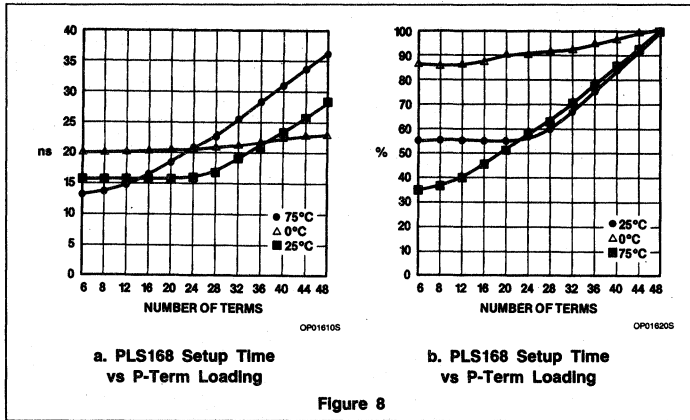
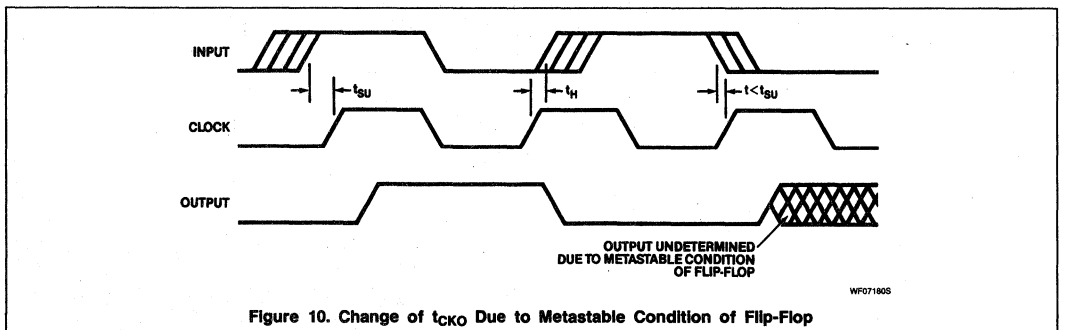
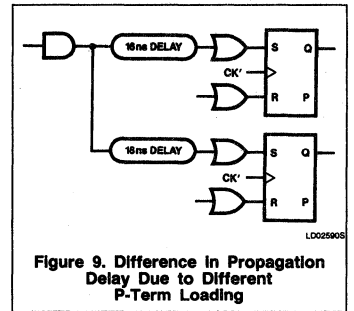


Figure 8



## PLS168/168A Primer

AN23

### Optional Preset/Output Enable

The PR/OE pin provides the user with the option of either using that pin to control the Tri-state output buffers of the Output Registers, or have that pin to asynchronously preset all registers to High. The purpose of the preset function is to provide the system a way to set the PLS168 to a known state, all Highs. The output enable function are sometimes used where the state machine is connected to a bus which is shared by other output circuits. It is also used during power-up sequence to keep the PLS168 from sending power glitches to other circuits which it drives. By programming the PR/OE pin to control the Tri-state output buffers, the preset function is permanently disabled. By programming the PR/OE pin to control the asynchronous preset of the registers, the output buffers are permanently enabled. While using the preset function to asynchronously preset the register, if a rising edge of the clock occurs while the preset input is High, the registers will remain preset. Normal flip-flop operation will resume only after the preset input is Low and the rising edge of the next clock. Setting the registers to a predefined pattern other than all Highs may be accomplished by using a dedicated p-term, which is activated by an input pin which will also inhibit all other p-terms which are being used. The inhibiting of other p-terms eliminates the problem of undetermined state of an RS flip-flop caused by having Highs on both R and S inputs.

### Diagnostic Features

In debugging a state machine, sometimes it is necessary to know what is the content of the state register. The buried State Register may be read by applying +10V on I<sub>0</sub>, which will cause the contents of register bits P<sub>4</sub> to P<sub>5</sub>, P<sub>6</sub> to P<sub>9</sub> to be displayed on output pins F<sub>2</sub> to F<sub>3</sub> and P<sub>0</sub> and P<sub>3</sub> respectively. While the device can handle the +10V on pin I<sub>0</sub>, prolonged and continuous use will cause the chip to heat up since more power is being dissipated at +10V. To facilitate more expedient functional tests, synchronous preset vectors as described above may be used to set the State Register to different states without having to go through the entire sequence.

### Timing Requirements

Since the PLS168 is intended to be a synchronous finite state machine, the inputs are expected to be synchronous to the clock and set-up and hold time requirements are expected to be met. In general, the set-up time requirement is measured at its worst case as having the entire AND-array connected to the OR-term being measured and there is only one active AND-term to drive the entire line. The set-up time decreases from there as less p-terms are used. This is due to the capacitance of the unused AND-terms being removed from the line. Figure 8a shows the typical set-up time requirement of a PLS168A device. Figure 8b shows the normalized set-up time as a percentage of the worst case,

which is with 48 terms connected. In a typical state machine design, some flip-flops will change states more frequently than others. Those that change more frequently will have more p-term loading on its OR gates than those that change states less frequently. The different loadings on the OR-terms cause different delay on the inputs of the flip-flops as shown in Figure 9. If an input fails to meet the set-up time specification, it is possible that the resultant of the input change gets to one set of flip-flops before the rising edge of the clock while it gets to other flip-flops during or after the clock's rising edge. The result is that some flip-flops have changed states and some have not, or some get into metastable condition as shown in Figure 10. The state machine is now either out of sequence or is in an undefined state. This problem often occurs with asynchronous inputs which is generated totally independent of the clock on the system. A common remedy for the problem of asynchronous inputs is to use latches or flip-flops to catch the input and then synchronously feed it to the state machine. This minimizes the problem with the different propagation delays due to different p-term loading. But there is still a finite probability that the external latches or flip-flops will get into metastable condition, which may be propagated into the state machine. Nevertheless, the window for the flip-flops in state machine to get into undefined states or metastable condition is narrowed by a great extent.

# PLS168/168A Primer

AN23

## APPENDIX A

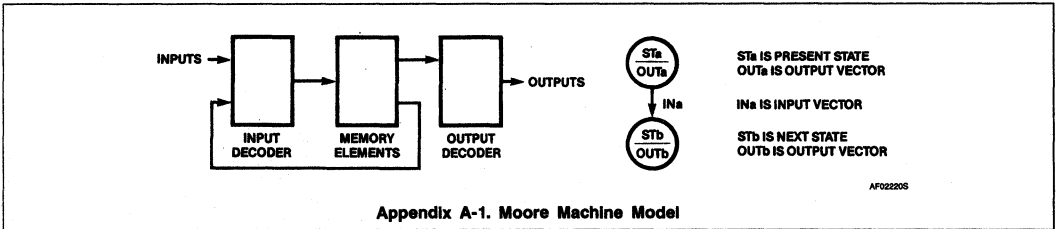
### INTRODUCTION TO STATE MACHINE

A state machine is a synchronous sequential circuit which interprets inputs and generates outputs in accordance with a predetermined logic sequence. It is analogous to running a computer program with a computer. The state machine, with its sequence coded in hard-

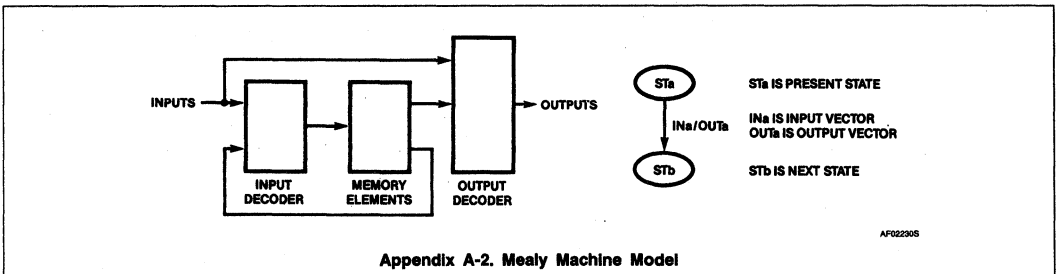
ware, can run much faster than a computer running the sequence in software. Therefore, it is often used in controller applications where speed is important.

Generally, state machines may be classified as Mealy or Moore machines as shown in Figures 1a and 1b. The fundamental difference of the two types are: the output of a Moore machine is a dependent of only the

state of the memory elements whereas the output of a Mealy machine is a dependent of both the state of the memory elements and the inputs to the state machine. The figures also show graphic representations of the logic sequence in the form of state diagram in which the bubbles represent state vectors, and the arrows represent transitions from present states to next states.



Appendix A-1. Moore Machine Model



Appendix A-2. Mealy Machine Model

# PLS168/168A Primer

# AN23

## APPENDIX B

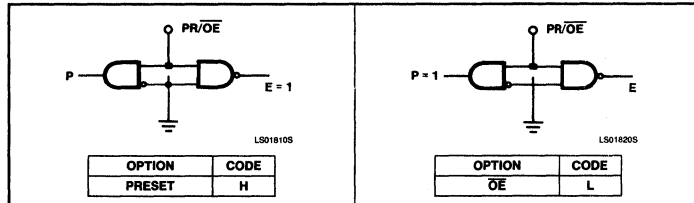
### LOGIC PROGRAMMING

The FPLS can be programmed by means of Logic programming equipment.

With Logic programming, the AND/OR gate input connections necessary to implement the desired logic function are coded directly from the State Diagram using the Program Table on the following page.

In this table, the logic state or action of control variables C, I, P, N, and F, associated with each Transition Term  $T_n$ , is assigned a symbol which results in the proper fusing pattern of corresponding link pairs, defined as follows:

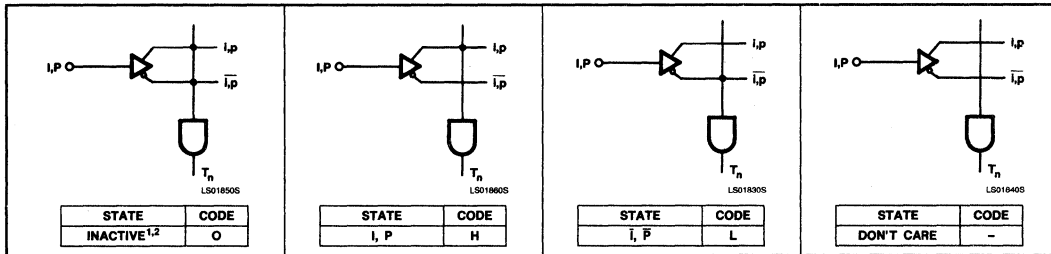
### PRESET/ $\overline{OE}$ OPTION - (P/E)



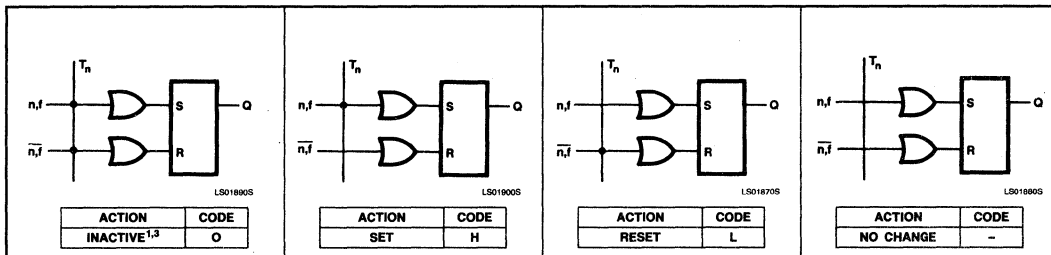
#### PROGRAMMING:

The PLS168A has a power-up preset feature. This feature insures that the device will power-up in a known state with all register elements (state and output register) at a logic high (H). When programming the device it is important to realize this is the initial state of the device. You *must* provide a next state jump if you do not wish to use all highs (H) as the present state.

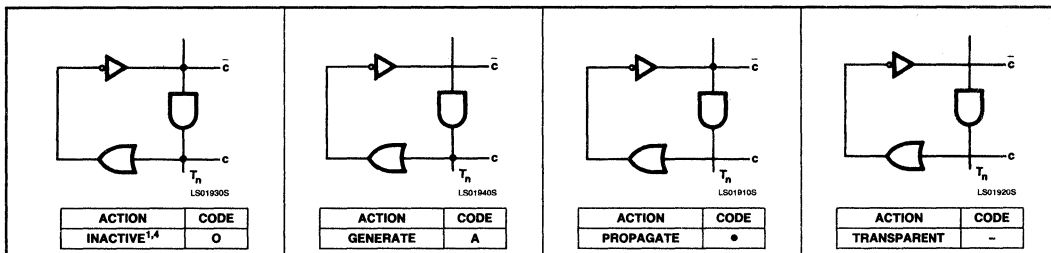
### "AND" ARRAY - (I), (P)



### "OR" ARRAY - (N), (F)



### "COMPLEMENT" ARRAY - (C)



#### NOTES:

1. This is the initial unprogrammed state of all link pairs. It is normally associated with all unused (inactive) AND gates  $T_n$ .
2. Any gate  $T_n$  will be conditionally inhibited if any one of its I or P link pairs is left intact.
3. To prevent simultaneous Set and Reset flip-flop commands, this state is not allowed for N and F link pairs coupled to active gates  $T_n$  (see flip-flop truth tables).
4. To prevent oscillations, this state is not allowed for C link pairs coupled to active gates  $T_n$ .

# AN24 PLS173 as a 10-Bit Comparator, 74LS460

## Application Note

### Application Specific Products

#### DESCRIPTION

The PLS173 is a 24-pin PLA device which has 10 bidirectional outputs and 12 dedicated inputs. The output of the device is the sum of products of the inputs. The polarity of each output may be individually programmed as Active-High or Active-Low. A logic diagram of the device is shown in Appendix A. A 10-bit comparator similar to the 74LS460 compares two 10-bit data inputs to establish if EQUIVALENCE or NOT EQUIVALENCE exists. The output has True and Complement comparison status outputs. The logic diagram of the comparator is shown in Figure 1.

The truth table is as shown in Table 1 where vectors a and b are 10-bit inputs to A9 to A0 and B9 to B0. If the input to A9-A0 is bit-to-bit equivalent to the input to B9-B0, the two input vectors are considered EQUIVALENT, and output EQ goes High and NE goes Low. If the two input vectors are not bit-to-bit equivalent, then EQ goes Low and NE goes High. The circuit is implemented with AMAZE as shown in Figures 2a, 2b and 2c. The result of logic simulation of the circuit is shown in Figure 2d.

Notice that on the OR side of the program table in Figure 5, all the fuses in the OR-term are intact, which means that all the AND-terms are still connected to all the OR-terms. This feature provides for future modification. But if all the unused AND-terms are deleted, the device will run faster. There are also many unused AND-terms which will provide for future modification. But if they are deleted (both on the AND and OR side), it will amount to about 450µA per term power saving. Figure 3 is the program table with all unused terms deleted.

Table 1. Function Table

A <sub>9</sub> - A <sub>0</sub>	B <sub>9</sub> - B <sub>0</sub>	EQ	NE
a	a	H	L
b	b	H	L
a	b	L	H
b	a	L	H

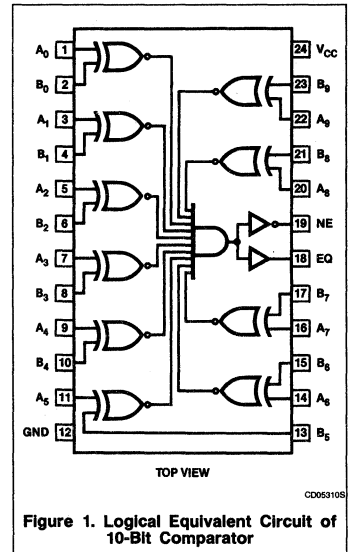
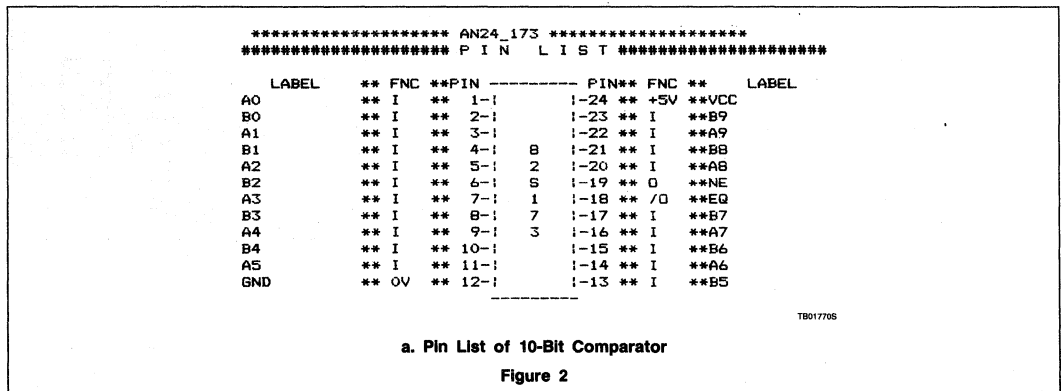


Figure 1. Logical Equivalent Circuit of 10-Bit Comparator



## PLS173 as a 10-Bit Comparator, 74LS460

AN24

```

***** AN24_173 *****
@DEVICE TYPE
@ZS173
@DRAWING
..... 10-BIT COMPARATOR USING PLS173
@REVISION
..... REV-0
@DATE
..... OCT-14-85
@SYMBOL
..... AN24_173
@COMPANY
..... SIGNETICS
@NAME
..... DAVID WONG
@DESCRIPTION
This circuit compares two 10-bit inputs. If they are bit-to-bit equivalent,
outputs EQ goes HIGH and NE goes LOW. If the inputs are not bit-to-bit equivalent
to each other, outputs EQ goes LOW and NE goes HIGH.
@COMMON PRODUCT TERM

T0 = A0 * /B0 ;
T1 = /A0 * B0 ;
T2 = A1 * /B1 ;
T3 = /A1 * B1 ;
T4 = A2 * /B2 ;
T5 = /A2 * B2 ;
T6 = A3 * /B3 ;
T7 = /A3 * B3 ;
T8 = A4 * /B4 ;
T9 = /A4 * B4 ;
T10 = A5 * /B5 ;
T11 = /A5 * B5 ;
T12 = A6 * /B6 ;
T13 = /A6 * B6 ;
T14 = A7 * /B7 ;
T15 = /A7 * B7 ;
T16 = A8 * /B8 ;
T17 = /A8 * B8 ;
T18 = A9 * /B9 ;
T19 = /A9 * B9 ;

@I/O DIRECTION
@OUTPUT POLARITY
@LOGIC EQUATION

EQ = / ( T0 + T1 + T2 + T3 + T4 + T5 + T6 + T7 + T8 + T9 +
        T10 + T11 + T12 + T13 + T14 + T15 + T16 + T17 + T18 + T19 ) ;

NE = T0 + T1 + T2 + T3 + T4 + T5 + T6 + T7 + T8 + T9 +
     T10 + T11 + T12 + T13 + T14 + T15 + T16 + T17 + T18 + T19 ;

```

TD017905

## b. Boolean Equations of 10-Bit Comparator

Figure 2 (Continued)

PLS173 as a 10-Bit Comparator, 74LS460

AN24

```

***** AN24_173 *****
Cust/Project - ..... DAVID WONG
Date        - ..... OCT-14-85
Rev/I. D.   - ..... REV-0

B2S173                                     ! POLARITY !
-----
T !                                         ! Hz Hz Hz Hz Hz Lz Hz Hz Hz !
E !-----
R !                                         ! B(i) ! B(o) !
M ! 1 1 -----
  ! 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 !
0 ! ----- L H ! ----- ! A A A A A A A A A A !
1 ! ----- H L ! ----- ! A A A A A A A A A A !
2 ! ----- L H ! ----- ! A A A A A A A A A A !
3 ! ----- H L ! ----- ! A A A A A A A A A A !
4 ! ----- L H ! ----- ! A A A A A A A A A A !
5 ! ----- H L ! ----- ! A A A A A A A A A A !
6 ! ----- L H ! ----- ! A A A A A A A A A A !
7 ! ----- H L ! ----- ! A A A A A A A A A A !
8 ! ----- L H ! ----- ! A A A A A A A A A A !
9 ! ----- H L ! ----- ! A A A A A A A A A A !
10 ! L H ----- ! A A A A A A A A A A !
11 ! H L ----- ! A A A A A A A A A A !
12 ! ----- L H ! ----- ! A A A A A A A A A A !
13 ! ----- H L ! ----- ! A A A A A A A A A A !
14 ! ----- L H ! ----- ! A A A A A A A A A A !
15 ! ----- H L ! ----- ! A A A A A A A A A A !
16 ! ----- L H ! ----- ! A A A A A A A A A A !
17 ! ----- H L ! ----- ! A A A A A A A A A A !
18 ! ----- L H ! ----- ! A A A A A A A A A A !
19 ! ----- H L ! ----- ! A A A A A A A A A A !
20 ! 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 !
21 ! 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 !
22 ! 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 !
23 ! 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 !
24 ! 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 !
25 ! 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 !
26 ! 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 !
27 ! 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 !
28 ! 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 !
29 ! 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 !
30 ! 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 !
31 ! 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 !
D9 ! 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 !
DB ! 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 !
D7 ! 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 !
D6 ! 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 !
D5 ! -----
D4 ! -----
D3 ! 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 !
D2 ! 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 !
D1 ! 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 !
D0 ! 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 !

  B A B A B A B A B A B A B A B A B A B A B A B A B A B A B A B A
  5 5 4 4 3 3 2 2 1 1 0 0 9 9 8 8 E 0 7 7 6 6 9 9 8 8 E 0 7 7 6 6

```

TBD17905

c. Program Table of 10-Bit Comparator After Assembly of Boolean Equation File  
Figure 2 (Continued)

PLS173 as a 10-Bit Comparator, 74LS460

AN24

```

B2S173  A:AN24_173.STD
" AMAZE FILE ID: AN24_173
" LOGIC SIMULATION OF 10-BIT COMPARATOR
"
" <==INPUTS==> <=B(I/O)=>  TRACE TERMS
" 11
" 109B76543210 9876543210
"
00000000000 0000LH0000 ;
01000000000 0000HL0000 ;
10000000000 0000HL0000 ;
11000000000 0000LH0000 ;
00010000000 0000HL0000 ;
00100000000 0000HL0000 ;
00110000000 0000LH0000 ;
00001000000 0000HL0000 ;
00001000000 0000HL0000 ;
00001100000 0000LH0000 ;
00000010000 0000HL0000 ;
00000010000 0000HL0000 ;
00000011000 0000LH0000 ;
00000000100 0000HL0000 ;
00000000100 0000HL0000 ;
000000001100 0000LH0000 ;
00000000001 0000HL0000 ;
000000000010 0000HL0000 ;
000000000011 0000LH0000 ;
000000000000 0100HL0000 ;
000000000000 1000HL0000 ;
000000000000 1100LH0000 ;
000000000000 0001HL0000 ;
000000000000 0010HL0000 ;
000000000000 0011LH0000 ;
000000000000 0000HL0100 ;
000000000000 0000HL1000 ;
000000000000 0000HL1100 ;
000000000000 0000HL0001 ;
000000000000 0000HL0010 ;
000000000000 0000LH0011 ;
"
" ----- I/O CONTROL LINES
"          IIII00IIII DESIGNATED I/O USAGE
"          IIII00IIII ACTUAL I/O USAGE
"
" PINLIST...
" 13 11 10 09 08 07 06 05 04 03 02 01
" 23 22 21 20 19 18 17 16 15 14 ;

```

TB016005

d. Test Vectors Generated by AMAZE After Logic Simulation  
Figure 2 (Continued)



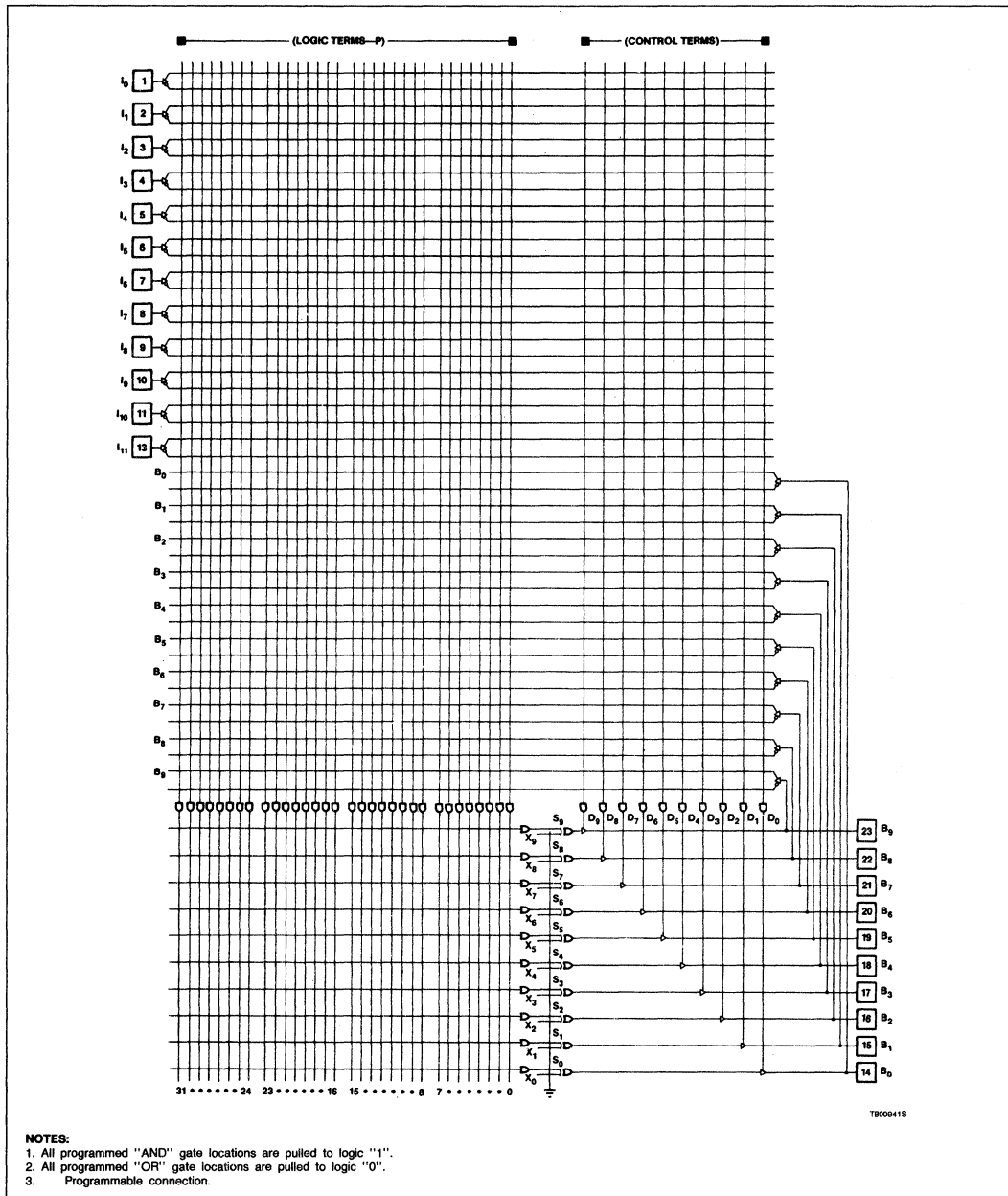


# PLS173 as a 10-Bit Comparator, 74LS460

AN24

## APPENDIX A

### FPLA LOGIC DIAGRAM FOR PLS173



# AN25

## A Simple Entry/Exit Metering Device

### Application Note

### Application Specific Products

#### DESCRIPTION

The following is the implementation of a counter/controller using a PLS157. This system counts and controls the number of entries to a certain media. For example, it can be used as a parking-lot controller to keep track of the number of cars entering a parking lot, and restricting any entries when the parking lot is full.

The design utilizes the six registers, six bi-directional I/O lines and the general architecture of the PLS157. In this design, the user can set the maximum number of entries using

external switches. Since the six registers in the device are used as a six-bit counter, the maximum number that can be set is  $2^6$ . However, the design can be cascaded to increase the number of entries.

#### SYSTEM OPERATION

The basic system configuration is shown in Figure 1. As can be seen from this figure, the system can be expanded indefinitely. All the devices in cascade are identical to each other. Whenever there is a need for expanding the maximum number of counts, a device

can be programmed identical to the device(s) already in the system and added in cascade.

There are external switches connected to the bi-directional inputs of the flip-flops. The user sets the maximum count in binary with the switches and loads the value into the registers. If the maximum count necessary exceeds the number available in one device, then another device is cascaded to the existing device. For example, in a system with two devices, the maximum available increases to  $2^{12}$  or 4096.

When the maximum count is set, the control circuitry is used to keep track of the number of entries and exits. In the case of a parking lot, a pressure sensor can be used at the entrance and exit to generate a pulse which triggers the control circuitry. An alternative could be light beam sensors to generate the necessary pulses.

The operation of the system is such that each entry decreases the maximum count stored in the registers by one. When the count is zero, it means that the media is full and the control circuitry generates a signal to disable any further entries to the system. On the other hand, each exit increases the count by one. Since it is assumed that the media is empty when the maximum count is set, it is, therefore, impossible to physically count past the maximum set number. This is due to the fact that there cannot be more exits from the system than the actual number of objects residing in the system. If there are objects in the system when the maximum number is set, these objects should also be taken into consideration when setting the maximum count.

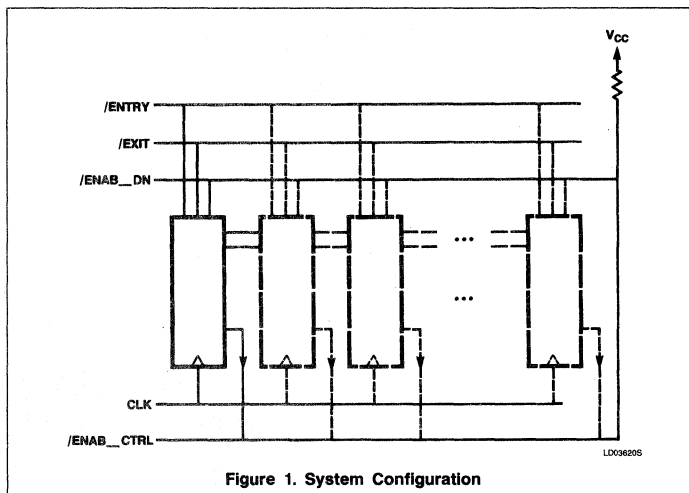


Figure 1. System Configuration

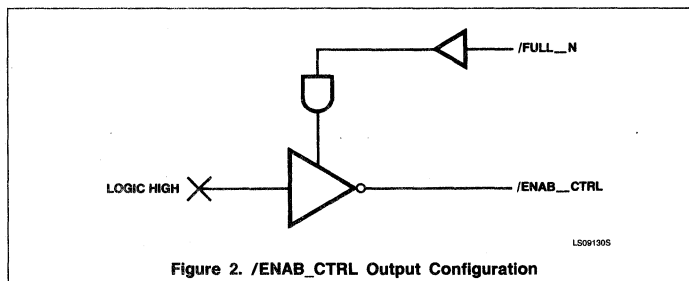


Figure 2. /ENAB\_CTRL Output Configuration

#### CONTROL SIGNALS

The following is the explanation of the control signals shown in Figure 1.

The /ENTRY and /EXIT inputs signify an entrance or exit to or from the respective media and are triggered from the sensors at the entrance and exit. Each entry causes a count-down from the maximum set number until all the flip-flops are low. This means that the system is full and the count-down should be disabled.

# A Simple Entry/Exit Metering Device

AN25

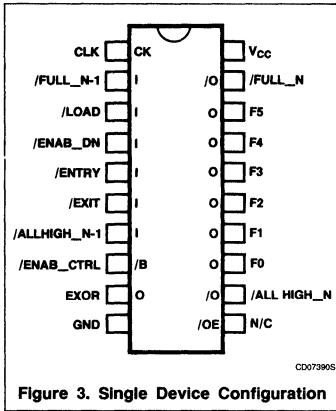


Figure 3. Single Device Configuration

When all the flip-flops are low, the /FULL\_N signal which is an Active-Low output is activated. The count-down is enabled by the /ENAB\_DN input, which is connected externally to the /ENAB\_CTRL line. The /ENAB\_CTRL is a Tri-state output which is controlled by the /FULL\_N signal. When all the flip-flops are low, the /FULL\_N signal deactivates the /ENAB\_CTRL output as shown in Figure 2.

All of the /ENAB\_CTRL outputs are connected to the /ENAB\_CTRL bus as shown in Figure 1. When the media is full, all the /ENAB\_CTRL signals are deactivated which cause the count-down to be disabled.

An exit causes a count up. As mentioned before, the total number of exits from the system cannot exceed the total number of entries.

The /ALLHIGH\_N signal is an Active-Low output which becomes low when all the flip-flops are high. Figure 3 shows the configuration of a single device.

The /FULL\_N and /ALLHIGH\_N signals are used to activate the /FULL\_N-1 and /ALLHIGH\_N-1 of a cascaded device. These

signals will be explained in detail shortly. The /FULL\_N signal is also used to trigger any control circuitry used to restrict any further entries to the system. In the case of a parking lot, the /FULL\_N can be used to activate a light or close the entrance gate.

The EXOR output is used to clock the system and is externally connected to the CLK input of the device. The EXOR generates a clock pulse when either an exit or entry take place. This restricts the system from counting if a simultaneous entry and exit takes place.

To set the maximum count, the user can use switches connected to inputs F0-F5 of the device(s). Then by bringing the /LOAD input low and triggering the /ENTRY or /EXIT inputs to generate a clock pulse, the data from these switches can be latched into the registers.

## CASCADE DEVICE EXAMPLE

This is an example of two cascaded devices. Figure 4 shows the cascade configuration of devices 'A' and 'B'. For the sake of simplicity, it is assumed that each device contains only two registers. This assumption gives clarity to the explanation, and can be extended to cover the six registers contained in each device.

As an example, it is assumed that the maximum count to be implemented in this example is 1001 in binary. In other words, the maximum number of entries is set to be 'ten'. F0A and F1A are the registers in device A, and F0B and F1B are the registers in device B. 10 is the input sequence to device B and 01 is input sequence to device A.

The count-up and count-down sequences are shown in Table 1.

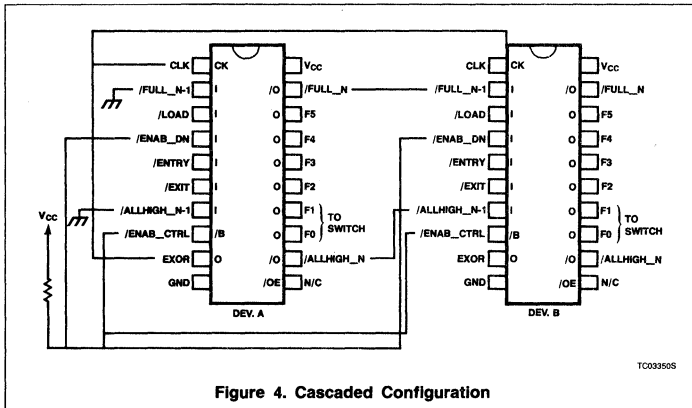


Figure 4. Cascaded Configuration

Table 1. An Up/Down Count Sequence Example

DEV. B		DEV. A		DEV. B		DEV. A	
F0B	F1B	F0A	F1A	F0B	F1B	F0A	F1A
1	0	0	1	0	0	0	0
0	0	1	0	0	0	0	1
0	1	1	1	0	0	1	0
0	0	0	1	0	1	0	1
0	1	0	0	0	0	1	0
0	0	1	1	0	1	1	0
0	0	1	0	0	1	1	1
0	0	0	1	1	0	0	0
0	0	0	0	1	0	0	1

COUNT DOWN

DEV. B		DEV. A		DEV. B		DEV. A	
F0B	F1B	F0A	F1A	F0B	F1B	F0A	F1A
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1
0	1	0	0	0	0	1	0
0	0	0	1	0	1	0	1
0	1	0	0	0	0	1	0
0	0	1	1	0	1	1	0
0	0	1	0	0	1	1	1
1	0	0	0	1	0	0	0
0	0	0	0	1	0	0	1

COUNT UP

Table 2. Count Sequence

NO.	F5	F4	F3	F2	F1	F0
0	0	0	0	0	0	0
1	0	0	0	0	0	1
2	0	0	0	0	1	0
3	0	0	0	0	1	1
4	0	0	0	1	0	0
...	...	...	...	...	...	...
7	0	0	0	1	1	0
8	0	0	0	1	0	1
...	...	...	...	...	...	...
15	0	0	1	0	0	0
16	0	0	1	0	1	0
...	...	...	...	...	...	...
31	0	1	0	0	0	0
32	0	1	0	1	0	0
...	...	...	...	...	...	...
62	1	1	1	1	1	0
63	1	1	1	1	1	1
...	...	...	...	...	...	...
0	0	0	0	0	0	0

# A Simple Entry/Exit Metering Device

AN25

In the 1001 example, F0B is the MSB (Most Significant Bit) and F1A is the LSB (Least Significant Bit). As can be seen in Table 1, a count-down occurs in device 'B', when F0A and F1A are both low. When F0A and F1A are low, this means that the /FULL\_N signal is active. The /FULL\_N-1 input of device 'B' is connected to the FULL\_N output of device 'A'. Therefore, a count down occurs in device 'B' only when the input to /FULL\_N-1 goes low. In this manner, the count-down sequence in Table 1 can be implemented.

Similarly, for the count-up sequence, a count-up occurs in device 'B' only when both F0A and F1A are high. At this point, the /ALLHIGH\_N signal becomes active. Therefore, a count-up occurs in device 'B' only when /ALLHIGH\_N of device 'A' is active.

## UP/DOWN COUNTERS

An up-down counter is implemented in each device to keep track of the number of entries and exits. To implement these counters the flip-flops are set in the toggle mode. Table 2 shows the count sequence.

The following conditions are true for the count-up sequence:

Table 3a

REGISTER	TRUE CONDITION FOR TOGGLE
F0	Unconditionally Toggle
F1	F0
F2	F0*F1
F3	F0*F1*F2
F4	F0*F1*F2*F3
F5	F0*F1*F2*F3*F4

Similarly, for the count-down sequence:

Table 3b

REGISTER	TRUE CONDITION FOR TOGGLE
F0	Unconditionally Toggle
F1	/F0
F2	/F0*/F1
F3	/F0*/F1*/F2*
F4	/F0*/F1*/F2*/F3
F5	/F0*/F1*/F2*/F3*/F4*

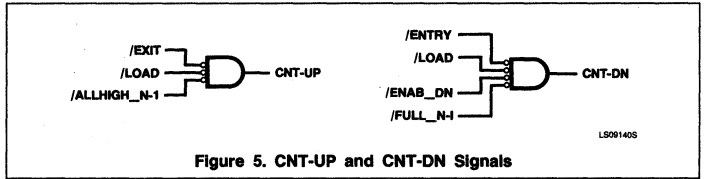


Figure 5. CNT-UP and CNT-DN Signals

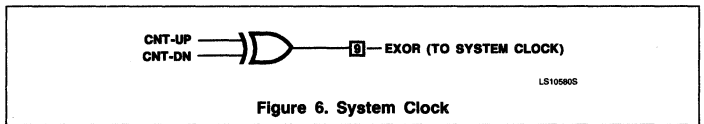


Figure 6. System Clock

```

File Name : METER
Date : 10/31/1985
Time : 14:35:31

***** P I N L I S T *****
      LABEL      ** FNC **PIN ----- PIN** FNC ** LABEL
CLK          ** CK ** 1-:      :-20 ** +5V **VCC
/FULL_N-1    ** I  ** 2-:      :-19 ** /0  **/FULL_N
/LOAD        ** I  ** 3-:      :-18 ** 0   **F5
/ENAB_DN     ** I  ** 4-:      P  :-17 ** 0   **F4
/ENTRY       ** I  ** 5-:      L  :-16 ** 0   **F3
/EXIT        ** I  ** 6-:      X  :-15 ** 0   **F2
/ALLHIGH_N-1** I  ** 7-:      1  :-14 ** 0   **F1
/ENAB_CTRL   ** /B ** 8-:      5  :-13 ** 0   **F0
EXOR         ** 0  ** 9-:      7  :-12 ** /0  **/ALLHIGH_N
GND          ** OV ** 10-:     :-11 ** /OE  **N/C
    
```

Figure 7

## CONTROL SIGNALS

The following are the control signals used to control the operation of the devices:

The logic diagram for generating the 'CNT-UP' and 'CNT-DN' signals is shown in Figure 5.

As can be seen from this figure, a count-up occurs when the following is true:

- a) An exit takes place and,
- b) the system is not loading in any data and,
- c) the previous stage has generated the /ALLHIGH\_N-1 signal.

A count-down occurs when:

- a) An entry takes place and,
- b) the system is not loading in any data and,
- c) the previous stage has generated the /FULL\_N-1 signal and,
- d) the media is not full, thus /ENAB\_DN signal is low.

The system clock is generated using the CNT-UP and CNT-DN signals. The system is

to be clocked only when an exit or entry takes place. In order to prevent the system from being clocked when a simultaneous entry and exit takes place, these two signals are exclusive-OR'ed (Figure 6).

The equations shown in (Tables 3a and 3b) are generated using Signetics AMAZE software (Figures 7 and 8). The resulting program table is shown in Table 4.

## A Simple Entry/Exit Metering Device

AN25

```

@DEVICE TYPE
PLX157
@DRAWING
@REVISION
@DATE
10-31-85

@SYMBOL
ENTR/EXIT METER

@COMPANY
SIGNETICS

@NAME
ALI GHEISSARI

@DESCRIPTION
*****
* The following is the implementation of a Entry/Exit Meter. A *
* application of this device would be in controlling the number of *
* entries to a certain media. An example is a parking lot controller *
* for restricting the number of cars entering the lot. *
* The maximum number is programmable and the design can be cacaded to *
* Accomodate an increase in the number of entries. *
* The system counts down from the maximum set number to zero *
* The number of exits cannot exceed the maximum count set by the *
* external switches,therefore the upper limit on the count will never *
* be exceeded. *
*****

@COMMON PRODUCT TERM
UPCNT = /LOAD * EXIT * ALLHIGH_N-1;    "/UPCNT is the signal generated for "
                                         "activating the count-up from the ENTRY "
                                         "input and the ALLHIGH signal from any "
                                         "connected devices "

DNCNT = /LOAD * ENTRY * FULL_N-1;      "DNCNT is similar to UPCNT except it is "
                                         "generated for the count down "

ALL-HIGH = /F0 * /F1 * /F2 * /F3 * /F4 * /F5;
ALL-LOW = F0 * F1 * F2 * F3 * F4 * F5;

@COMPLEMENT ARRAY
/C = /(0);

@I/O DIRECTION
D0 = 0;                                "/EXIT is an input "
D1 = 0;                                "ALLHIGH_N-1 is an input "
D2 = /FULL_N;                          "When maximum count is reached /FULL_N "
                                         "disables /ENAB-CTRL which in turn disables "
                                         "the device(s) from counting down "
                                         "/ENAB-CTRL is connected to the ENAB_DN of the "
                                         "device(s) "
D3 = 1;                                "EXQR is an output "
D4 = 1;                                "/ALLHIGH_N is an output "
D5 = 1;                                "/FULL_N is an output "

```

TB019306

Figure 8

## A Simple Entry/Exit Metering Device

AN25

```

@OUTPUT POLARITY
X2 = 0;           "/ENAB_CTRL is active low           "
X3 = 1;           "EXDR is active high             "
X4 = 0;           "/ALLHIGH_N is active low        "
X5 = 0;           "/FULL_N is active low           "

@FLIP FLOP CONTROL
FC = 1;           "All flip flops are J-K             "

@OUTPUT ENABLE
"NOT USED"

@REGISTER LOAD
LA = LOAD;        "Load the registers with the number set   "
LB = LOAD;        "by the input switches                     "

@ASYNCHRONOUS PRESET/RESET
"NOT USED"

@FLIP FLOP MODE
"NOT USED"

@LOGIC EQUATION
EXDR = UPCNT*/DNCNT + DNCNT*/UPCNT; "Output count pulse only if there is not  "
                                     "a simultaneous entry and exit          "
/FULL_N = /( ALL-LOW * /LOAD);      "indicates when maximum count is reached "
/ALLHIGH_N = /( ALL-HIGH * /LOAD);  "shows when empty                       "
/ENAB_CTRL = /(1);                  "puts low on the ENAB_CTRL bus         "

"The following is the implementation of the counter together with the  "
"necessary logic."

/F0: T = (ENAB_DN * DNCNT) + UPCNT;

/F1: T = (ENAB_DN * F0 * DNCNT) + (/F0 * UPCNT);

/F2: T = (ENAB_DN * F0 * F1 * DNCNT) + (/F0 * /F1 * UPCNT);

/F3: T = (ENAB_DN * F0 * F1 * F2 * DNCNT) +
(/F0 * /F1 * /F2 * UPCNT);

/F4: T = (ENAB_DN * F0 * F1 * F2 * F3 * DNCNT) +
(/F0 * /F1 * /F2 * /F3 * UPCNT);

/F5: T = (ENAB_DN * F0 * F1 * F2 * F3 * F4 * DNCNT) +
(/F0 * /F1 * /F2 * /F3 * /F4 * UPCNT);

```

TB019405

Figure 8 (Continued)

# A Simple Entry/Exit Metering Device

# AN25

Table 4. Program Table for Entry/Exit Meter

```

Cust/Project - ALI GHEISSARI
Date - 10-31-85
Rev/I. D. -

PLX157
-----
T | | | | | F/F TYPE | EB | EA | | | POLARITY |
R | | | | | A:A:A:A:A:A | 0 | 0 | | | L:L:H:L:L:L |
M |-----|-----|-----|-----|-----|-----|-----|
C | 1 | 3 | 2 | 1 | 0 | 1 | 5 | 4 | 3 | 2 | 1 | 0 | 1 | 5 | 4 | 3 | 2 | 1 | 0 |
0! | - | H | H | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
1! | - | H | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
2! | - | L | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
3! | - | L | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
4! | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
5! | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
6! | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
7! | - | L | H | L | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
8! | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
9! | - | L | H | L | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
10! | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
11! | - | L | H | L | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
12! | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
13! | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
14! | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
15! | - | L | H | L | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
16! | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
17! | - | L | H | L | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
18! | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
19! | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
20! | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
21! | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
22! | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
23! | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
24! | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
25! | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
26! | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
27! | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
28! | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
29! | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
30! | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
31! | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
Fc! | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
Pb! | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
Rb! | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
Lb! | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
La! | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
D5! | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
D4! | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
D3! | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
D2! | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
D1! | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
D0! | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

/ / / / / E / / / / F F F F F F F F F F F F F F / E / / / /
F E E E L X E A F A S 4 3 2 1 0 S 4 3 2 1 0 L X E A F A
U N N X D D N I U L U O N L U L U O N L U L
L A T I A R A L L L A R A L L L
L B R T D B H L H D B H L H
- Y I I I I I I I I
N D C G N G C G N G
- N T H H T H H
I R R R R R R R R R
L N N L N N
I I

T801950S
    
```



### Application Specific Products

#### FEATURES

- 100% functional replacement for all 20-pin PALs
- I/O propagation delay: 15ns (max)
- Security fuse lock
- 10 inputs
- 8 bidirectional I/O lines
- Tri-state outputs have programmable polarity
- Architecture: 8 groups of nine AND gates. Total of 72 product terms
- Software support on Signetics AMAZE
- Complete TTL compatibility
- Each bidirectional I/O has individually controllable output enable

#### ARCHITECTURE

The PLHS18P8A is an oxide-isolated, bipolar field-programmable logic array. This device is configured as a decoding two-level AND-OR (sum of products) structure. The PLHS18P8A block diagram is shown in Figure 1. All the AND gates are linked to ten inputs ( $I_0 - I_9$ ) and eight bidirectional I/O lines ( $B_{(0)} - B_{(7)}$ ). These links can be made via the on-chip true/complement buffers. The 72 AND gates are configured in 8 groups which contain 9 AND gates each. In every group, eight AND gates are used for user-defined logic functions and the ninth AND gate is used as a tri-state output enable control. This gives the user capability to control the output enable by means of a product term. The outputs of the eight logical product terms are ORed together (see Figure 1). The output polarity of each OR gate is individually programmable via an Exclusive-OR gate. The user has a choice of Active-Low or Active-High on each of the eight outputs. Figure 2 shows the logic diagram of the PLHS18P8A.

#### HOW A DEVICE IS PROGRAMMED

There are three main programmable sections on the PLHS18P8A:

- A. The AND array.
- B. The output polarity.
- C. The security fuse.

**A. The AND array** — The AND array fuses are back-to-back diode pairs which act as open connections in a virgin device. These open connections are configured as 'Don't

Cares' ('-') in an unprogrammed device. The open connections are pulled to a logic high ('1') (see Figure 3). Consequently, all unprogrammed AND locations are pulled to a logic high ('1') state. This means that in an unprogrammed device, all the product terms are active. During fusing, current is avalanched across individual diode pairs. This essentially short circuits the diode and provides a connection for the associated product term. Figure 3 shows how a typical connection is made to the AND array (see Appendix B for a description of the vertical fuses).

The inputs to the AND array consist of 10 dedicated inputs ( $I_0 - I_9$ ) and 8 bidirectional I/Os ( $B_{(0)} - B_{(7)}$ ). Initially, all these inputs are configured as 'Don't Cares' ('-'). These inputs can be connected to the AND array through an inverting or non-inverting buffer. The AND gate can be connected to the inverting buffer by programming the inverting fuse. Similarly, a connection can be made to the non-inverting buffer by programming the non-inverting fuse. Disabling an AND term is achieved by implementing a logical low ('0') on the output of the specified AND gate. At least one pair of fuses must be programmed for one input to disable the product term. When both fuses are programmed, both the inverting and non-inverting buffers are connected to the inputs of the AND gate. This will in turn create a logical Low ('0') on the output.

In each block of nine AND gates, one gate is used as an output enable control for the tri-state output. The remaining eight gates are connected to a fixed OR gate. Since in the unprogrammed state the outputs of all the AND gates are at logic High ('1'), the output of the OR gate also acquires a logic High state. Therefore the user is responsible for deactivating any unwanted product terms. This is done by creating a logic Low ('0') on the outputs of the unwanted product terms as previously explained. Moreover, the output buffer is always enabled since the product term controlling the tri-state output buffer also has a logic High state. If any of the bidirectional pins are to be defined as inputs, the product term controlling the specified bidirectional pin must be disabled. The bidirectional pin can also be configured as a dynamic I/O by defining the required logic for the output-enable-control product term.

**B. The Output Polarity** — In an unprogrammed device, all the output polarities are configured as inverting buffers. The outputs of the PLHS18P8A are configured as tri-state buffers. The two inputs of each Ex-OR gate

are connected as follows: one input is connected to the output of the fixed OR gate. The second input is a connection to ground (logic low) through a fusible link. Figure 4 shows how an Ex-OR gate is used to define the output polarity.

As mentioned earlier, an unprogrammed fuse acts as an open connection which is pulled to a logic high. Therefore the output of the Ex-OR gate acts as an inverting buffer. When the fuse is programmed, there is a connection between ground (logic Low) and the input of the Ex-OR gate. This will cause the Ex-OR gate to act as a non-inverting buffer.

**C. The Security Fuse** — Programming equipment used to program the PLHS18P8A are capable of determining the logic pattern stored in this device (see Appendix A). The security fuse can be blown to disable the programmer from reading the pattern in a programmed device. This feature adds a measure of protection for proprietary designs.

The procedure for programming this fuse depends on the programmer manufacturer and is explained in the manufacturer's operations manual.

#### DESIGN TOOLS FOR THE PLHS18P8A

Many CAD tools such as Signetics' AMAZE software<sup>1</sup> are available to implement designs using the PLHS18P8A. The AMAZE software enables the user to enter the design in the form of Boolean equations or via the program table shown in Table 1. This program table is a one-to-one map of all the programmable links of Figure 2. The following explains the implementation of designs using the program table. Also, an example on using the Boolean Logic Entry program of the AMAZE software is given.

**Using PTE (Program Table Entry)** — Assume that Z is a typical logic function with the following equation:

$$Z = P_0 + P_1$$

Where  $P_0$  and  $P_1$  are product terms with the following equations:

$$P_0 = A * /B * C$$

$$P_1 = /A * B * /D$$

#### NOTE:

1. Refer to Signetics Programmable Logic Data Manual for a complete description of the AMAZE programmable logic development software.

PLHS18P8A Primer

AN26

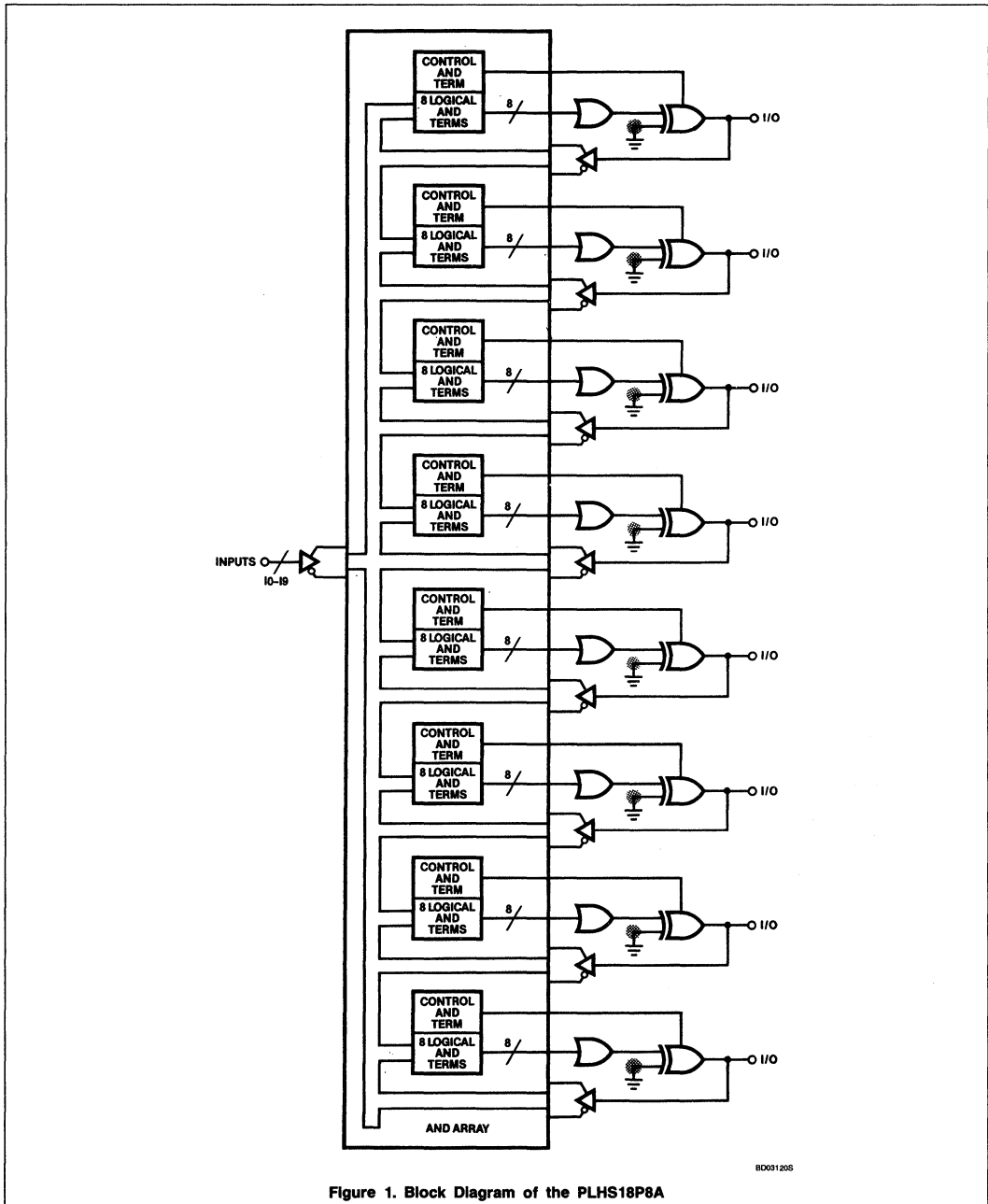
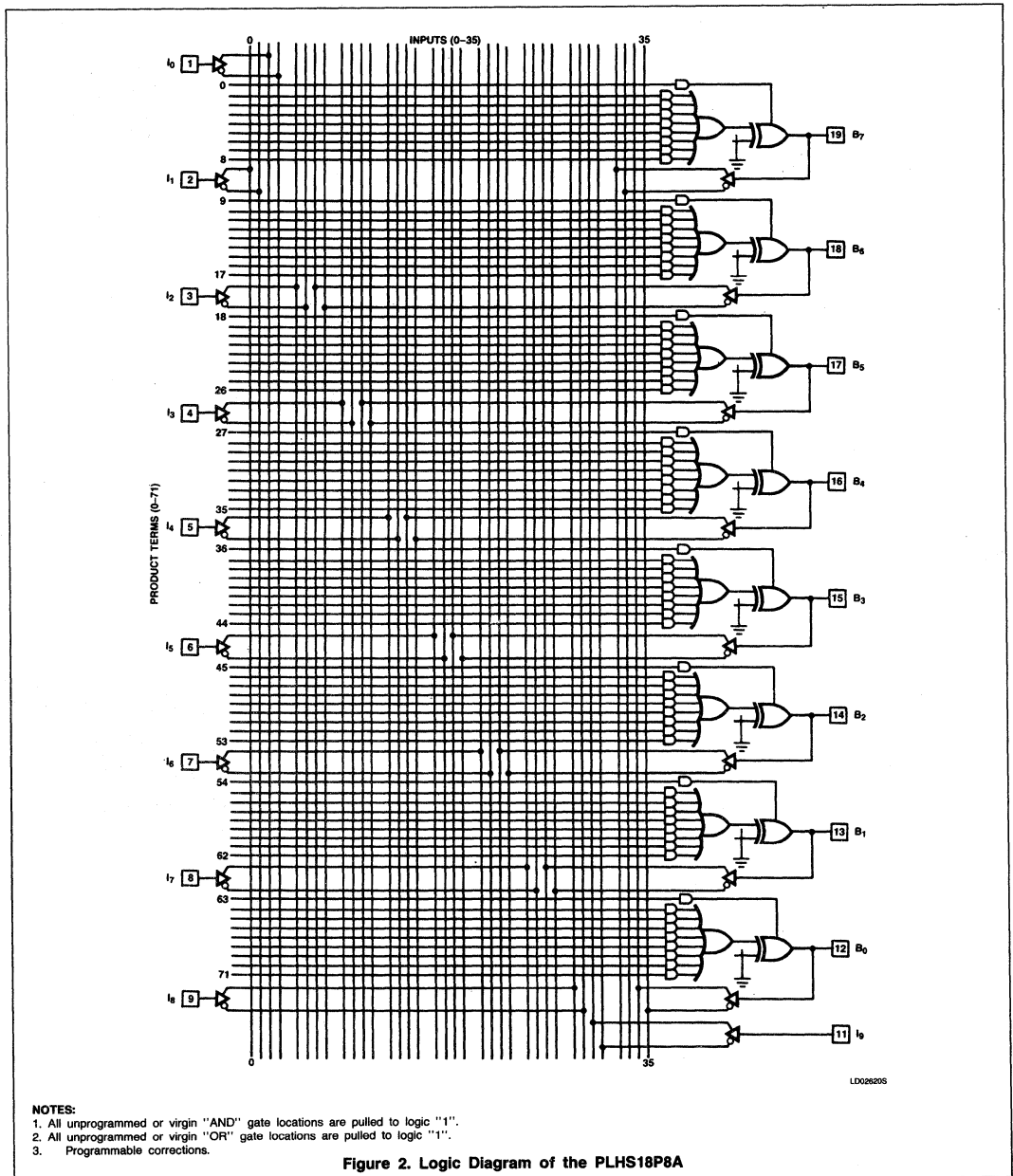


Figure 1. Block Diagram of the PLHS18P8A

PLHS18P8A Primer

AN26



# PLHS18P8A Primer

# AN26

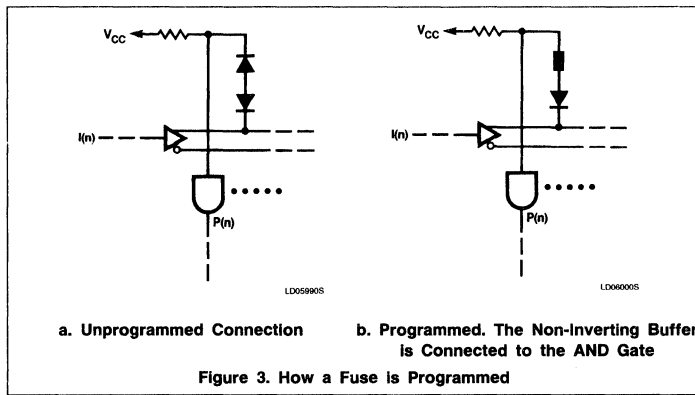


Figure 3. How a Fuse is Programmed

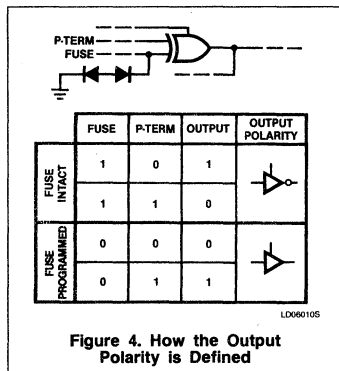


Figure 4. How the Output Polarity is Defined

The program table in Table 1 is used to implement this equation. Table 2 shows the implementation of the logic function using the PTE. The first group of AND terms is used to implement this function. B(7) is used for the output, I0, I1, and I2 as inputs.

- Term 0 is the direction control term. It is the tri-state output enable control term. Since Z is configured as an output, leaving Term 0 in its unprogrammed state causes the output to be unconditionally enabled.
- Term 1 is the P-term labeled P0, where  $A^*/B^*C$  is designated by HLH in columns I0, I1, and I2.

- Term 2 is the P-term labeled P1, where  $/A^*B^*/D$  is designated by LHL in columns I0, I1, and I2.
- Terms 3, 4, 5, 6, 7. At least one input (or all the inputs) must be set to zero to disconnect these terms from the OR gate.
- The output polarity for B(7) is H, and this is entered in the 'Polarity' section.

**Using BLAST (Boolean Logic And State Transfer)** — The BLAST module in AMAZE can be used to implement the above equation. Figures 5 and 6 show the pin list and logic equation format. Table 3 is the program table generated from these equations.

## APPENDIX A Programmiers

The PLHS18P8A can be programmed by means of logic programming equipment. With Logic Programming, the AND/Ex-OR gate input connections necessary to implement the desired logic are coded directly from the logic equations using the program table shown in Table 1.

The symbols used in the program table correspond to the fusing pattern of the corresponding link pairs, defined in Figure 6.

To program the device, the address of every fuse to be programmed must be entered in the programmer. This is a tedious and error-

prone method of implementing the required logic pattern. Using CAD software, such as Signetics' AMAZE software, enables users to go directly from program table or logic equations to a fuse plot. The fuse plot can be downloaded to a programmer through a serial port.

## APPENDIX B

The vertical fuse is the latest in programmable logic technology. It combines reliability, low capacitance and testability in an incredibly small space.

The vertical fuse takes advantage of the properties of silicon and aluminum. The virgin fuse is a three-layer device: a shallow layer of N-type silicon on top, a layer of P-type silicon in the middle, and a layer of N-type silicon on the bottom. This forms a pair of PN diodes, back-to-back, which will not allow current to pass under normal circumstances.

There is a cap of aluminum on top of this structure. During programming, high current conditions (50 to 100 times what is seen during normal operation) is induced by avalanche breakdown of the reverse-biased diode. The aluminum will "spike" through the shallow N-type layer. Once the aluminum has spiked through, the top diode in the pair is shorted out. The whole vertical fuse will then look like a pure PN diode. Before programming, the fuse is an excellent blocking element, having current leakages in the order of nanoamps. After programming, the fuse is set as a small, well-defined diode.

## QUALITY

Besides the AC and C parametrics, each and every fuse goes through three tests for forward characteristics, reverse characteristics, and programmability. Extensive on-chip test circuitry ensures full AC parametric testing before and after the part is programmed. This insures that the customer receives the highest possible fusing yield which is made possible by vertical fuses. Vertical fuses also offer the smallest and fastest array structure together with the highest reliability possible.

# PLHS18P8A Primer

# AN26

		POLARITY																											
		OR (FIXED)																											
		B (O)																											
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0												
<b>Notes</b> 1. The FPLA is shipped with all links open. 2. Unused I and B bits in the AND array exist as Don't Care. 3. All p-terms are active until programmed otherwise. 4. Unused p-terms must be disabled or programmed with at least one pair of fuses.	<b>CONTROL</b> (POL) HIGH L LOW L	<b>AND</b> O H L INACTIVE I, B I, B DON'T CARE	<b>T E R M</b>	<b>AND</b>																									
				<b>I</b>								<b>B (I)</b>																	
				9	8	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0								
			0																			D							
			1																			A							
			2																			A							
			3																			A							
			4																			A							
			5																			A							
			6																			A							
			7																			A							
			8																			A							
			9																			A							
			10																			A							
			11																			A							
			12																			A							
			13																			A							
14																			A										
15																			A										
16																			A										
17																			A										
18																			D										
19																			A										
20																			A										
21																			A										
22																			A										
23																			A										
24																			A										
25																			A										
26																			A										
27																			A										
28																			D										
29																			A										
30																			A										
31																			A										
32																			A										
33																			A										
34																			A										
35																			A										
36																			D										
37																			A										
38																			A										
39																			A										
40																			A										
41																			A										
42																			A										
43																			A										
44																			A										
45																			A										
46																			D										
47																			A										
48																			A										
49																			A										
50																			A										
51																			A										
52																			A										
53																			A										
54																			D										
55																			A										
56																			A										
57																			A										
58																			A										
59																			A										
60																			A										
61																			A										
62																			A										
63																			A										
64																			D										
65																			A										
66																			A										
67																			A										
68																			A										
69																			A										
70																			A										
71																			A										
PIN	11	9	8	7	6	5	4	3	2	1	19	18	17	16	15	14	13	12	19	18	17	16	15	14	13	12			
VARIABLE NAME																													

Table 1. Program Table

T802020S



PLHS18P8A Primer

AN26

```

File Name : 18PRIMER
Date : 5/22/1986
Time : 13:52:54

***** P I N L I S T *****

  LABEL  ** FNC **PIN  ----- PIN** FNC ** LABEL
A      ** I ** 1-1  P  1-20 ** +SV **VCC
B      ** I ** 2-1  L  1-19 ** 0  **Z
C      ** I ** 3-1  H  1-18 ** /0 **INVERT
D      ** I ** 4-1  S  1-17 ** I  **INPUT
N/C    ** I ** 5-1  1  1-16 ** B  **BIDIR
N/C    ** I ** 6-1  S  1-15 ** /B **/BDIR
N/C    ** I ** 7-1  P  1-14 ** /B **N/C
N/C    ** I ** 8-1  S  1-13 ** /B **N/C
N/C    ** I ** 9-1  A  1-12 ** /B **N/C
GND    ** OV ** 10-1 ----- 1-11 ** I  **N/C
    
```

Figure 5

```

File Name : 18PRIMER
Date : 5/22/1986
Time : 13:53:59

@DEVICE TYPE
PLHS18P8A
@DRAWING
..... PLHS18P8A PRIMER

@REVISION
..... PLHS18P8A REV '-'

@DATE
..... May 1986

@SYMBOL
..... PLHS18P8A

@COMPANY
..... SIGNETICS

@NAME
..... Ali Sheisari

@DESCRIPTION
*****
* This is an example of implementing equations for
* the PLHS18P8A using AMAZE
*****

@COMMON PRODUCT TERM
P0 = A * /B * C; " P0 and P1 are used to declare common
P1 = /A * /B * /D; " product terms.

@I/O DIRECTION
D7 = 1; " Defines D7 to be an output
D6 = 1; " Defines D6 to be an output
D5 = 0; " Defines D5 to be an input
D4 = A; " D4 an output when A is true otherwise an input
D3 = B; " D3 an output when B is true otherwise an input

@OUTPUT POLARITY
X7 = 0; " Documents the output polarity as active high
X6 = 1; " Documents the output polarity as active high
X4 = 0; " Documents the output polarity as active high
X3 = 1; " Documents the output polarity as active high

@LOGIC EQUATION
"Examples of some simple equations
Z = P0 + P1;

INVERT = /( INPUT );
BIDIR = /BDIR * B ;
/BDIR = /( A * C * D * /Z * INVERT * INPUT * BIDIR );
    
```

Figure 6. Boolean Equation File

# PLHS18P8A Primer

# AN26

File Name : 18PRIMER  
Date : 5/22/1986  
Time : 13:56:13

Cust/Project - ..... All Whaisari  
Date - ..... May 1986  
Rev/I. D. - ..... PLHS18P8A REV '...'

PLHS18P8A		POLARITY	
T	:	!HsLsHsLsLsL!	
E	:	I	B(t)
N	:		B(o)
0:	0:	0:	0:
1:	0:	0:	0:
2:	0:	0:	0:
3:	0:	0:	0:
4:	0:	0:	0:
5:	0:	0:	0:
6:	0:	0:	0:
7:	0:	0:	0:
8:	0:	0:	0:
9:	0:	0:	0:
10:	0:	0:	0:
11:	0:	0:	0:
12:	0:	0:	0:
13:	0:	0:	0:
14:	0:	0:	0:
15:	0:	0:	0:
16:	0:	0:	0:
17:	0:	0:	0:
18:	0:	0:	0:
19:	0:	0:	0:
20:	0:	0:	0:
21:	0:	0:	0:
22:	0:	0:	0:
23:	0:	0:	0:
24:	0:	0:	0:
25:	0:	0:	0:
26:	0:	0:	0:
27:	0:	0:	0:
28:	0:	0:	0:
29:	0:	0:	0:
30:	0:	0:	0:
31:	0:	0:	0:
32:	0:	0:	0:
33:	0:	0:	0:
34:	0:	0:	0:
35:	0:	0:	0:
36:	0:	0:	0:
37:	0:	0:	0:
38:	0:	0:	0:
39:	0:	0:	0:
40:	0:	0:	0:
41:	0:	0:	0:
42:	0:	0:	0:
43:	0:	0:	0:
44:	0:	0:	0:
45:	0:	0:	0:
46:	0:	0:	0:
47:	0:	0:	0:
48:	0:	0:	0:
49:	0:	0:	0:
50:	0:	0:	0:
51:	0:	0:	0:
52:	0:	0:	0:
53:	0:	0:	0:
54:	0:	0:	0:
55:	0:	0:	0:
56:	0:	0:	0:
57:	0:	0:	0:
58:	0:	0:	0:
59:	0:	0:	0:
60:	0:	0:	0:
61:	0:	0:	0:
62:	0:	0:	0:
63:	0:	0:	0:
64:	0:	0:	0:
65:	0:	0:	0:
66:	0:	0:	0:
67:	0:	0:	0:
68:	0:	0:	0:
69:	0:	0:	0:
70:	0:	0:	0:
71:	0:	0:	0:

NNNNNND CBAZ IIB/NNZ IIB/NNN  
 / / / / / NNIB / / / NNIB / / /  
 CCCCCC V P D D C C C V P D D C C C  
 E U I I E U I I  
 R T R R R T R R  
 T T

T900815

Table 3. Programming Table Generated by AMAZE





### Application Specific Products

#### INTRODUCTION

The PLHS473 is a 24-pin field-programmable logic array (FPLA) which has 11 dedicated inputs, 2 dedicated outputs and 9 bidirectional I/Os. The logic array consists of 24 programmable product terms which are connected to 22 programmable OR terms in the classical PLA architecture. A functional diagram is shown in Figure 1. The shaded areas represent programmable interconnects between vertical and horizontal lines. Eleven dedicated inputs,  $I_0$  to  $I_{10}$ , are located on the upper lefthand corner of the diagram, and the 9 bidirectional I/Os,  $B_0$  to  $B_8$ , are located on the lower righthand corner. All inputs to the AND array have true and inverting input buffers. The output portion of the 9 bidirectional I/Os and 2 outputs may be programmed to be Active-High or Active-Low by altering fuses  $X_0$  to  $X_8$ ,  $X_A$  and  $X_B$  which are connected to one leg of the Exclusive-OR gates. Each output is connected to two OR-terms, one of which provides the logic function OR to the output while the other provides the ENABLE function for the tri-state output buffer. Alternatively, each output may be configured to emulate an open-collector output by programming the output to an unconditional LOW and apply the logic function on the tri-state controlling OR-term as shown in Figure 2. In addition, the PLHS473 has a security fuse which may be programmed to lock out unauthorized access to the fuse map of the design.

This device is fabricated with an oxide-isolation process for the best speed/power performance. The programmable element is a "vertical fuse" which is actually two PN diodes connected back-to-back (anode-to-anode

stacked on top of one another) as shown in the insets of Figure 1. The fuses are normally open-circuited. To program a fuse, a sufficiently high voltage is applied across the two diodes such that one diode breaks down which induces metal migration across the PN junction of the avalanched diode. This technology allows smaller chip size and faster speed performance. The guaranteed propagation delay may be tested by using the two factory-programmed test columns and a test row as a test vehicle for speed testing. These test columns are to be deleted automatically in the process of programming.

The vertical fuses of an unprogrammed device, being normally open-circuited, set all AND terms to a High state and disconnected to all OR terms. The OR terms are normally Low when they are not connected to the AND terms. Once an AND term is programmed to make connections with the OR array, it cannot be disconnected. However, AND terms may be disabled by having the true and inverting input buffers (e.g.,  $A^* / A$ ,  $B^* / B$ , etc.) of all inputs connected to the AND term which is to be disabled.

While even a single input, ( $A^* / A$ ), will hold the AND term to a Low, glitches may develop if the input voltage changes state. Therefore, it is recommended that the complementary buffers of all inputs in the disabled AND term be connected.

#### Programming the PLHS473

The programming table and symbols of the PLHS473 are shown in Table 1 where there is a place for every function that is shown in Figure 1. The table is divided into two main

sections. The left side of the table, sections A and B, represents the input side of the AND array (section A has all dedicated inputs  $I_0$  to  $I_{10}$ , section B has inputs from the bidirectional I/Os  $B_0$  to  $B_8$ ), whereas the right side, sections C, D, and E, represents the outputs of the OR array. Section C represents the OR array which controls the tri-state output buffers of the bidirectional I/Os  $B_0$  to  $B_8$  and outputs  $O_A$  and  $O_B$ , while section D represents the OR array which drives the outputs. Section E controls the output polarity of each output. As shown in Table 1, the program table is very similar to a truth table. Each row represents one of the 24 product terms, while each column in sections A and B represents an input to the 24 product terms and in sections C and D, each column represents an output. The intersection of a column and a row in sections A and B represents four possible fuse configurations, namely, 1) both inverting and non-inverting input buffers are connected to the AND term, 2) only the inverting buffer is connected, 3) only the non-inverting input buffer is connected, and 4) both inverting and non-inverting buffers are disconnected. These four configurations are represented as "0", "L", "H", and "--", respectively. Each intersection of a row and a column in sections C and D involves only two fuse configurations, either connected or disconnected. Therefore, the entry there is either "A" or "--", respectively. In the polarity control section (section E), each square represents the configuration of the fuse connected to the Exclusive-OR, which is connected to a particular output pin. An "H" represents an Active-High or non-inverting output, an "L" represents an Active-Low or inverting output. See Table 1 for further details.

PLHS473 Primer

AN27

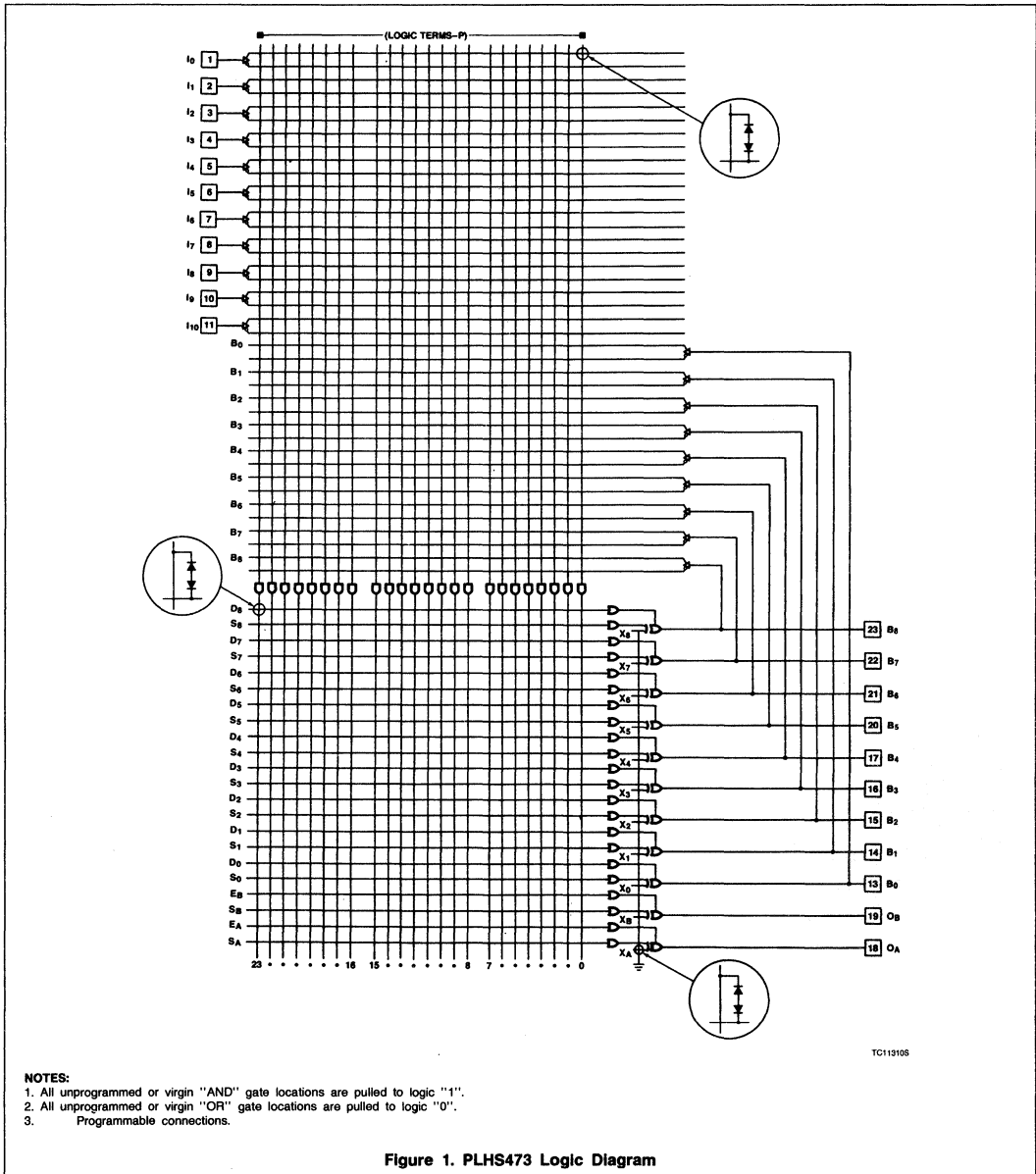


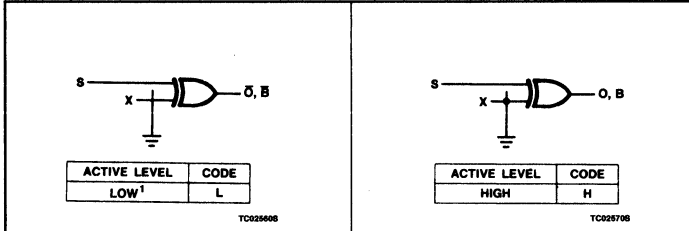
Figure 1. PLHS473 Logic Diagram



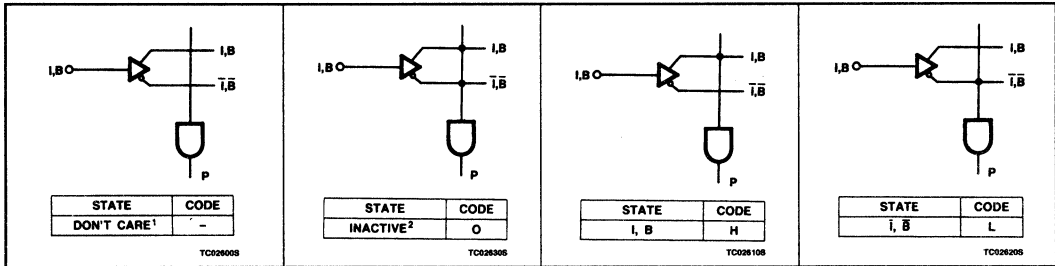
PLHS473 Primer

AN27

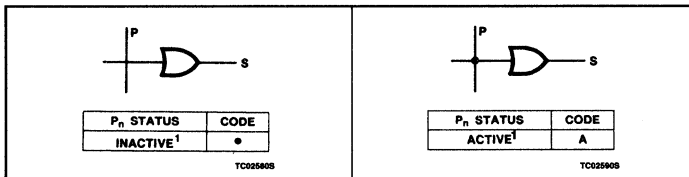
OUTPUT POLARITY — (O, B)



AND ARRAY — (I, B)



OR ARRAY — (O, B)



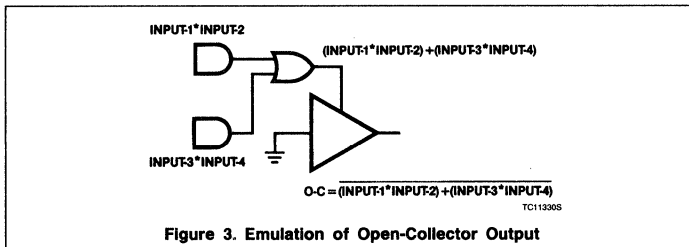
VIRGIN STATE

A factor shipped virgin device contains all fusible links.

1. All output at "L" polarity.
2. All P<sub>n</sub> terms are enable.
3. All P<sub>n</sub> terms are active on all outputs.

NOTES:

1. This is the initial unprogrammed state of all link pairs.
2. Any gate P<sub>n</sub> will be unconditionally inhibited if any one of its (I, B) link pairs is programmed for a connection.



PLHS473 Primer

AN27

Once we have determined the outputs, we can proceed with defining the logic functions. As an example, the logic function

OUT-0 = INPUT-1 \* INPUT-2

is shown in Table 2, term #2 where in the column labeled "INPUT-1", an "H" is entered to represent a connection of the non-inverting input buffer of INPUT-1 and the AND term 2. In the column labelled "INPUT-2", an "L" is entered to represent the connection of the inverting input buffer of INPUT-1 and the AND term 2. On the output side of the table, O\_B in term 0 is programmed "A" to make a connection between term 2 and the OR term which is connected to the output pin O\_B. The output polarity of O\_B is arbitrarily set to Active-High by programming an "H" in the O\_B column of the polarity section.

A sum of several product terms (AND-OR) is implemented by connecting multiple AND terms to the same OR term. An example is shown in Table 2, terms 4 and 5, which implement the logic function

OUT-1 = (INPUT-1 \* INPUT-2) + (INPUT-3 \* INPUT-4)

The logic function

OUT-2 = 1/((INPUT-1 \* INPUT-2) + (INPUT-3 \* INPUT-4))

is shown in terms 9 and 10. The output is Active-Low as programmed "L" in the polarity section of OUT-2. As a rule, the AND function is implemented in a row, whereas the OR function is implemented in a column. Since the AND portions of terms 2 and 4 are the same, they may be combined as shown in term 7, which saves a duplicated term.

Table 2 terms 12 and 13 together emulate an open-collector output as an example. The logic equation

O-C = 1/((INPUT-1 \* INPUT-2) + (INPUT-3 \* INPUT-4))

is implemented by first programming O-C Active-High in the polarity section. Terms 12 and 13 implement the sum of product func-

tion. Notice that the B(O) of O-C is programmed as a "dot" in both terms 12 and 13, which, together with the polarity control, causes it to be unconditionally Low, since a disconnected OR term is unconditionally Low. The intersections of column D\_7 and terms 12 and 13 are programmed "A" so that the tri-state output buffer of B(O)\_7 is enabled at input conditions (INPUT-1 \* INPUT-2) = 1 or (INPUT-3 \* INPUT-4) = 1. Terms 10 and 11 may be represented as shown in Figure 3.

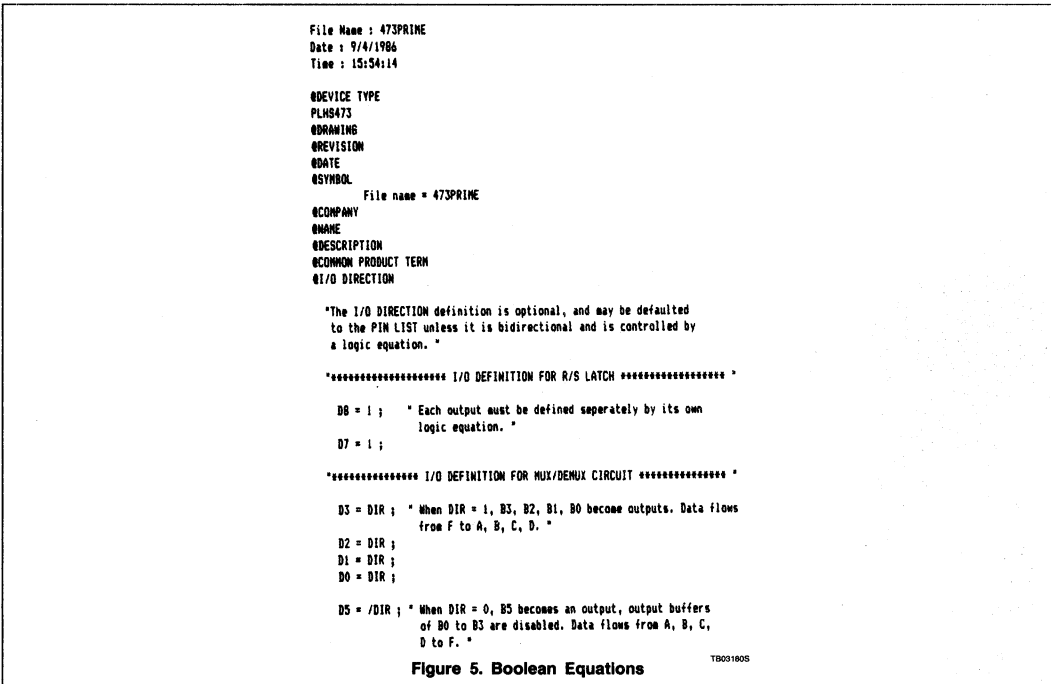
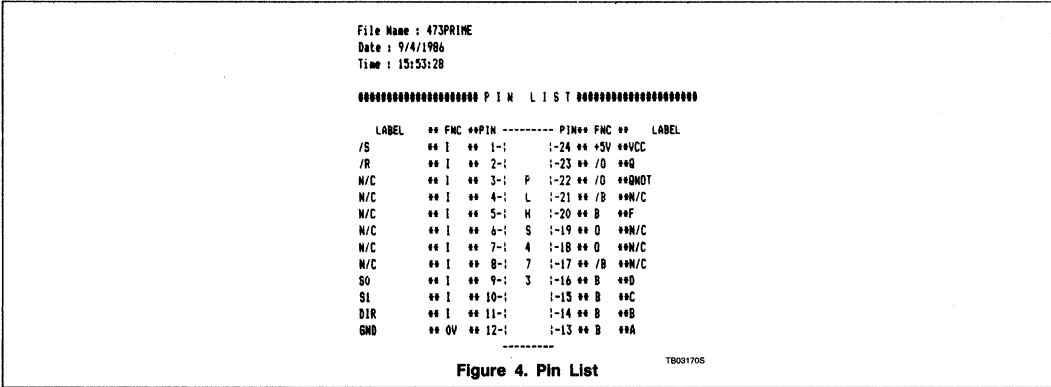
Signetics produces a software package, AMAZE, as a design tool which assists implementation of logic design and documentation. For further information on AMAZE software, refer to the AMAZE User's Manual. Two simple circuits shown as examples are implemented using AMAZE. The first circuit is a simple cross-coupled RS latch. The second circuit is a multiplexer/demultiplexer which multiplexes four inputs to one output or demultiplexes one input into four outputs. Both circuits are put into the same PLHS473 device.

Table 2. Examples of Programming PLHS473

Table 2. Examples of Programming PLHS473. Includes customer information form, logic diagrams for AND/OR terms, and a detailed programming grid with columns for terms (0-23) and pins (11-13), and rows for inputs (A, B, C, D), outputs (A, B, C, D), and polarity.

PLHS473 Primer

AN27



## PLHS473 Primer

AN27

**@OUTPUT POLARITY**

\* The definition of output polarity is optional and may be defaulted the PIN LIST. \*

\* \*\*\*\*\* POLARITY DEFINITION FOR R/S LATCH \*\*\*\*\* \*

X8, X7 = 1 ; \* Outputs with the same polarity may be defined in the same equation. Since the outputs Q and QNOT are defined as active LOW, this equation causes the XOR to function as an inverter. \*

\* \*\*\*\*\* POLARITY DEFINITION FOR MUX/DEMUX CIRCUIT \*\*\*\*\* \*

X5, X3, X2, X1, X0 = 0 ; \* Outputs B0, B1, B2, B3, B5 are active HIGH. \*

**@LOGIC EQUATION**

\* \*\*\*\*\* EQUATIONS FOR R/S LATCH \*\*\*\*\* \*

Q = /( /S \* QNOT ) ; \* The equation must used the format output = /( ..... ) ; since output Q is defined as active LOW. \*

QNOT = /( /R \* Q ) ;

\* \*\*\*\*\* EQUATIONS FOR MUX/DEMUX CIRCUIT \*\*\*\*\* \*

F = ( A \* /S0 \* /S1 ) +  
 ( B \* S0 \* /S1 ) +  
 ( C \* /S0 \* S1 ) +  
 ( D \* S0 \* S1 ) ;

A = ( F \* /S0 \* /S1 ) ;

B = ( F \* S0 \* /S1 ) ;

C = ( F \* /S0 \* S1 ) ;

D = ( F \* S0 \* S1 ) ;

**Figure 5. Boolean Equations (Continued)**

TB031905

PLHS473 Primer

AN27

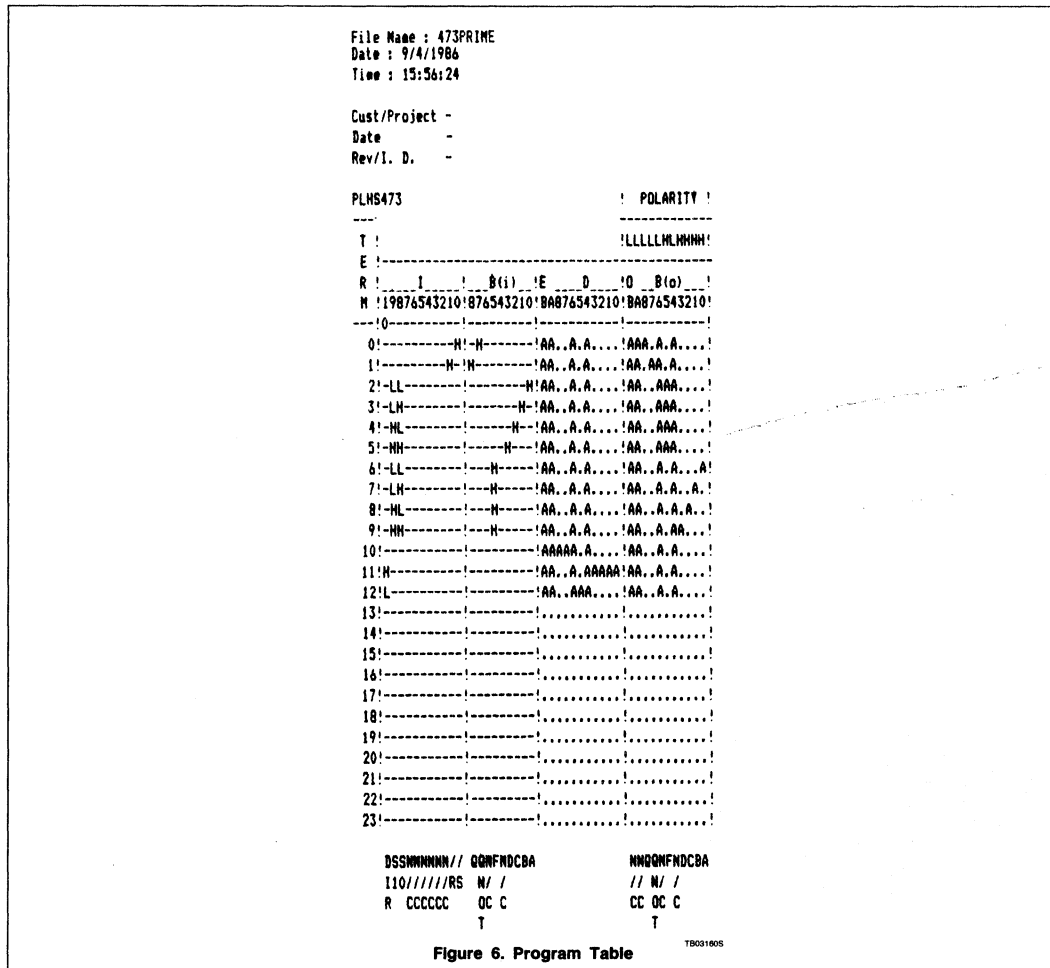


Figure 6. Program Table



## Application Specific Products

### INTRODUCTION

The general technique underlying the operation of this A/D converter is illustrated by the functional block diagram in Figure 1. The system consists of a D/A converter, a comparator circuit, and digital logic circuitry. The digital logic circuitry outputs a digital value which is converted to analog by the D/A converter.

The comparator senses when the output is greater or less than the input and causes the digital circuit to decrement or increment its digital output respectively. The initial conversion is completed in 13 clock cycles. If tracking mode is used, the A/D converter then tracks the input voltage as it changes by incrementing or decrementing 1-LSB per clock. The tracking function makes it possible to make an A/D conversion in one clock cycle if the input changes less than the value of 1-LSB per clock period. The conversion may be halted and the digital output, as well as the converted analog output from DAC, will hold their output constant indefinitely. This feature works well as sample-and-hold since its output voltage will not decay over time whereas the output of an analog sample/hold will decay due to charge leakages.

In order to avoid the violation of setup time by the output of the comparator, its output is latched. There is a built-in 2-phase clock in U2 which may be used to drive the logic circuitry and the latch of the comparator (see Signetics NE5105 data sheet for details on output latches of voltage comparators).

The analog input voltage may be sampled and held by an analog sample/hold circuit to

keep the input to the ADC from changing. The DONE output may be used to control the sample-and-hold if needed.

This paper discusses only the digital circuit which contains the SAR and the Up/Down Counter. The analog circuits are not within the scope of this paper.

### SAR

Two PLS179s are connected together to form a 12-bit shift register and up/down counter. The schematic diagram of the A/D converter is shown in Figure 2. U2 contains bits 0 to 4 and U1 contains Bits 5 to 11. Interconnects are made as shown in the diagram. The digital output to the DAC is in natural binary format (e.g. 0000 0000 0000 equal zero, and 1111 1111 1111 is full scale or 4095). After the /ST input becomes 0, at the rising edge of the next clock, the SAR is initialized to half-scale (1000 0000 0000) and the DONE flip-flop is reset to output 0 which causes the open-collector output /DONE\_OC to become high impedance. The digital output is converted by the DAC and is compared to the analog input voltage by the comparator. If the digital output is greater than the analog input, the SAR shifts the 1 to next MSB on the right. The content of the SAR becomes (0100 0000 0000). If the digital output is still greater than the input, the SAR shifts right one bit again. The content of the SAR then becomes (0010 0000 0000). The shifting of 1 to the next MSB is equivalent to reducing by half the value of the bit under consideration. If the output is still too large, the SAR reduces it by half again by shifting to the right one more time.

The SAR keeps shifting to the right until the digital output is less than the input. When the output is less than the input, the SAR adds one bit to the next MSB while keeping all the higher order bits unchanged. For example, if the current output is 0001 0000 0000 and the output is less than the input, the SAR adds one bit to the right at the next clock. The output becomes 0001 1000 0000. The output is again compared to the input. If the addition of that one bit is too much, it will be shifted to the right until the output becomes less than the input. When that happens, that SAR will again add one bit to the right. The algorithm of the SAR may be summarized as the following: If the output is greater than the input, shift to the right; otherwise add one bit to the right. This process continues until all 12 bits have been operated on. The last bit (Bit 0) is always changed from 0 to 1, which is used as the condition to set DONE to 1 which, in turn, sets open-collector output /DONE\_OC, to 0.

### UP/DOWN COUNTER

After DONE becomes 1, if /ST and /HOLD are 1 and /TRACK is 0, the SAR turns into a 12-bit up/down counter. If the analog input voltage increases, the counter will increment by 1 at every clock until it matches the input. If the input decreases, the counter will decrement by 1. When /HOLD becomes 0, the counter is inhibited and the output is held indefinitely. The counters consist of 12 toggle flip-flops and 2 p-terms per flip-flop for directional control. The counter will operate only

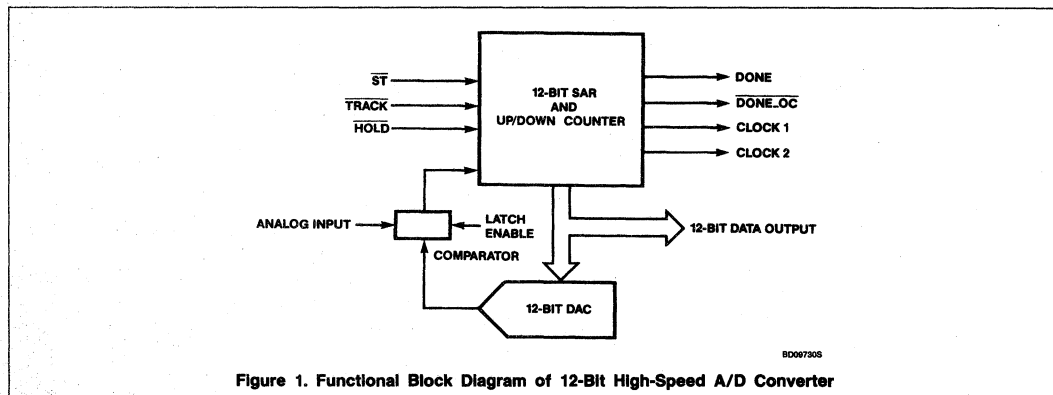


Figure 1. Functional Block Diagram of 12-Bit High-Speed A/D Converter

# High-Speed 12-Bit Tracking A/D Converter Using PLS179

AN28

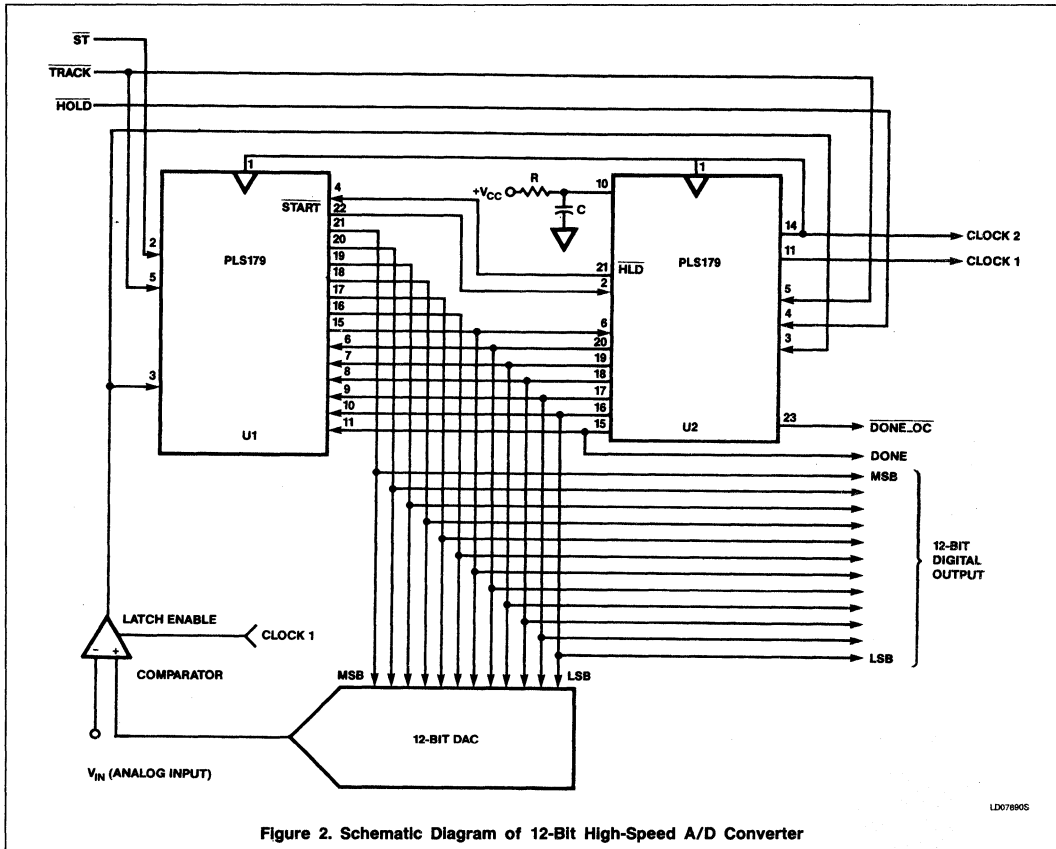


Figure 2. Schematic Diagram of 12-Bit High-Speed A/D Converter

LD07890S

# High-Speed 12-Bit Tracking A/D Converter Using PLS179

## AN28

after the approximation cycle is completed and DONE is 1.

Since the /ST and /HOLD inputs may be asynchronous with the clock, in order to minimize the possibilities of having a metastable condition from happening, these inputs close-up are latched by flip-flops /START of U1 and /HLD of U2 respectively. Once they are latched, subsequent operation begins at the rising-edge of the next clock. The output of the comparator may be latched to prevent setup time violation. (Signetics NE5105 is a high-speed comparator with an output latch. External latch may be used with other comparators.)

### CLOCKS

U2 generates an optional 2-phase clock which may be used to control the latch of the comparator. The two clocks are basically 180° out of phase and CLOCK2 has an additional 25ns propagation delay behind CLOCK1. CLOCK2 is used to drive the clock-inputs of the PLS179 devices.

The clock frequency is controlled by R and C. Those who want to use the built-in clock

should experiment with RC time constants for the best value. It is recommended that the capacitance should be less than 1000pF for best results (see Ap Note AN13 for more detail).

### DONE AND /DONE\_OC

The output DONE is reset to 0 when /ST is 0. It remains 0 until the approximation cycle is completed. After the least significant bit becomes 1, the DONE bit becomes 1 at the next clock. It remains 1 until it is reset again by input /ST.

The /DONE\_OC output is configured to emulate an open-collector output. The output is programmed to have a logic 0. When DONE is 0, the Tri-state output buffer is set to Hi-Z condition. As soon as DONE equals 1, the Tri-state buffer is enabled and /DONE\_OC becomes 0.

In the initial phase of A/D conversion, 13 clock cycles are required. It is essential that the input voltage to the comparator remains unchanged while the SAR is converting. It may be necessary to have a sample/hold at

the front end. The DONE output may be used to control the analog sample/hold circuit.

### INPUT LATCHES

Flip-flop /START and 2 p-terms in U1 are configured as a non-inverting D flip-flop. The input, /ST, and the output /START have the same polarities. Flip-flop /HLD and 2 p-terms in U2 also form a non-inverting D flip-flop. The output /HLD and the input /HOLD have the same polarities.

### AMAZE IMPLEMENTATION

The implementation of the logic circuit using AMAZE is as shown in the appendices. The SAR circuit is first designed as a state machine (file name: ADCS.SEE). It is then partitioned into two PLS179s after proper pin assignments are made. Then the up/down counter, input latches, 2-phase clocks and the open-collector output, are implemented by using Boolean equations in their respective .BEE files (file names: ADCB1.BEE and ADCB2.BEE) in AMAZE. The files are then assembled to produce the fuse-maps of PLS179 (ADCB1.STD and ADCB2.STD).

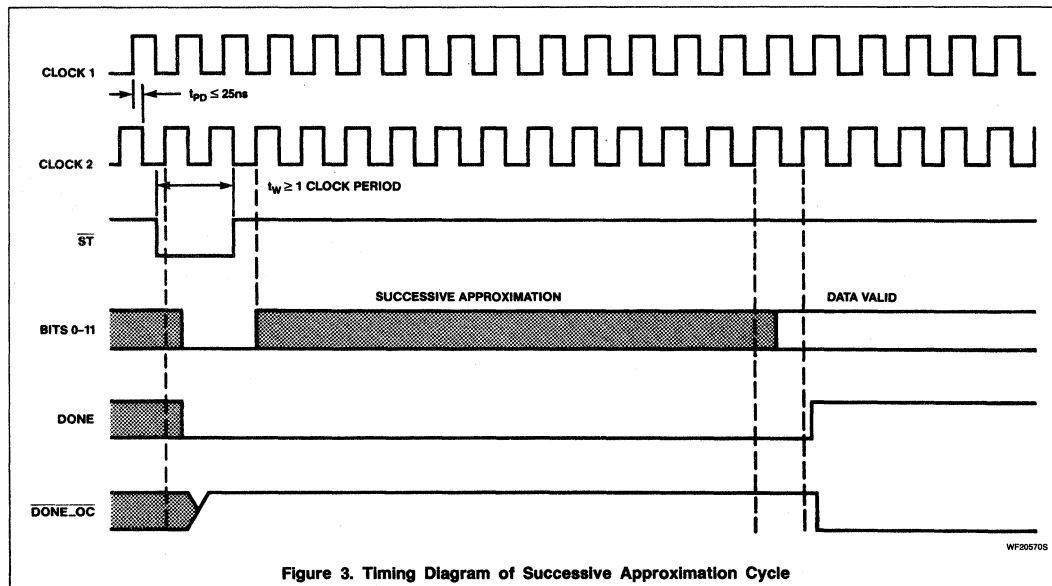


Figure 3. Timing Diagram of Successive Approximation Cycle

WF205705

## High-Speed 12-Bit Tracking A/D Converter Using PLS179

AN28

## APPENDIX A: STATE EQUATIONS OF SAR

```

File Name : ADCS
Date : 10/21/1986
Time : 11:2:14

@DEVICE SELECTION
ADC1/PLS179
ADC2/PLS179

@STATE VECTORS
[ /START, BIT11, BIT10, BIT9, BIT8, BIT7, BIT6, BIT5, BIT4, BIT3, BIT2,
  BIT1, BIT0, DONE ]

INIT          = 0 ---- - - - - - b ; "START CONVERSION PROCESS"
HALFSCALE.   = - 1000 0000 0000 0 b ; "SET SAR TO HALF SCALE"
ST2048       = 1 1000 0000 0000 0 b ; "PRESENT STATE = 2048 (HALF SCALE)"
ST1024       = 1 -100 0000 0000 0 b ;
ST512        = 1 ---10 0000 0000 0 b ;
ST256        = 1 ---1 0000 0000 0 b ;
ST128        = 1 ---- 1000 0000 0 b ;
ST64         = 1 ---- -100 0000 0 b ;
ST32         = 1 ---- --10 0000 0 b ;
ST16         = 1 ---- ---1 0000 0 b ;
ST8          = 1 ---- ---- 1000 0 b ;
ST4          = 1 ---- ---- -100 0 b ;
ST2          = 1 ---- ---- --10 0 b ;
ST1          = 1 ---- ---- ---1 0 b ;

AD1024       = - -1-- - - - - - b ; "ADD 1 BIT TO THE RIGHT"
AD512        = - - -1- - - - - b ;
AD256        = - - ---1 - - - - - b ;
AD128        = - - ---- 1-- - - - - b ;
AD64         = - - ---- -1- - - - - b ;
AD32         = - - ---- --1- - - - - b ;
AD16         = - - ---- ---1 - - - - b ;
AD8          = - - ---- ---- 1-- - - - b ;
AD4          = - - ---- ---- -1- - - - b ;
AD2          = - - ---- ---- --1- - - - b ;
AD1          = - - ---- ---- ---1 - - - b ;
END          = - - ---- ---- --- 1 b ;

SH1024       = - - 01-- - - - - - b ; "SHIFT ONE BIT TO THE RIGHT"
SH512        = - - -01- - - - - - b ;
SH256        = - - --01 - - - - - b ;
SH128        = - - --0 1-- - - - - - b ;
SH64         = - - ---- 01-- - - - - - b ;
SH32         = - - ---- --01- - - - - b ;
SH16         = - - ---- ---01- - - - - b ;
SH8          = - - ---- ---- 0 1-- - - - b ;
SH4          = - - ---- ---- 01-- - - - b ;
SH2          = - - ---- ---- --01- - - b ;
SH1          = - - ---- ---- ---01- - - b ;
SH0          = - - ---- ---- ---0 1 b ;

@INPUT VECTORS
[ COMPARE ]
GREATER = 1 b ; "IF DIGITAL OUTPUT IS GREATER THAN ANALOG INPUT, ...."
LESS    = 0 b ; "IF DIGITAL OUTPUT IS LESS THAN ANALOG INPUT, ...."

@OUTPUT VECTORS

```

T8033205

## High-Speed 12-Bit Tracking A/D Converter Using PLS179

AN28

## APPENDIX A: STATE EQUATIONS OF SAR (Continued)

```

@TRANSITIONS
WHILE [ INIT ]
  IF [ ] THEN [ HALFSCALE ] "INITIALIZE REGISTER TO HALF SCALE"
WHILE [ ST2048 ]
  IF [ GREATER ] THEN [ SH1024 ] "IF GREATER THAN, SHIFT 1 BIT"
  IF [ LESS ] THEN [ AD1024 ] "IF LESS THAN, ADD 1 BIT"
WHILE [ ST1024 ]
  IF [ GREATER ] THEN [ SH512 ]
  IF [ LESS ] THEN [ AD512 ]
WHILE [ ST512 ]
  IF [ GREATER ] THEN [ SH256 ]
  IF [ LESS ] THEN [ AD256 ]
WHILE [ ST256 ]
  IF [ GREATER ] THEN [ SH128 ]
  IF [ LESS ] THEN [ AD128 ]
WHILE [ ST128 ]
  IF [ GREATER ] THEN [ SH64 ]
  IF [ LESS ] THEN [ AD64 ]
WHILE [ ST64 ]
  IF [ GREATER ] THEN [ SH32 ]
  IF [ LESS ] THEN [ AD32 ]
WHILE [ ST32 ]
  IF [ GREATER ] THEN [ SH16 ]
  IF [ LESS ] THEN [ AD16 ]
WHILE [ ST16 ]
  IF [ GREATER ] THEN [ SH8 ]
  IF [ LESS ] THEN [ AD8 ]
WHILE [ ST8 ]
  IF [ GREATER ] THEN [ SH4 ]
  IF [ LESS ] THEN [ AD4 ]
WHILE [ ST4 ]
  IF [ GREATER ] THEN [ SH2 ]
  IF [ LESS ] THEN [ AD2 ]
WHILE [ ST2 ]
  IF [ GREATER ] THEN [ SH1 ]
  IF [ LESS ] THEN [ AD1 ]
WHILE [ ST1 ]
  IF [ GREATER ] THEN [ SH0 ]
  IF [ ] THEN [ END ]

```

TB003005

## High-Speed 12-Bit Tracking A/D Converter Using PLS179

AN28

## APPENDIX B: PIN LISTS

File Name : ADCB2  
Date : 10/21/1986  
Time : 10:58:26

\*\*\*\*\* P I N L I S T \*\*\*\*\*

LABEL	** FNC	**PIN	-----	PIN**	FNC **	LABEL
CLOCK	** CK	** 1-!		!-24	** +5V	**VCC
/START	** I	** 2-!		!-23	** /B	**/DONE_OC
COMPARE	** I	** 3-!		!-22	** B	**N/C
/HOLD	** I	** 4-!	P	!-21	** 0	**/HLD
/TRACK	** I	** 5-!	L	!-20	** 0	**BIT4
BIT5	** I	** 6-!	S	!-19	** 0	**BIT3
N/C	** I	** 7-!	1	!-18	** 0	**BIT2
N/C	** I	** 8-!	7	!-17	** 0	**BIT1
N/C	** I	** 9-!	9	!-16	** 0	**BIT0
RC	** /B	** 10-!		!-15	** 0	**DONE
CLOCK1	** 0	** 11-!		!-14	** /0	**CLOCK2
GND	** OV	** 12-!		!-13	** /OE	**N/C

TB033405

File Name : ADCB1  
Date : 10/21/1986  
Time : 10:53:17

\*\*\*\*\* P I N L I S T \*\*\*\*\*

LABEL	** FNC	**PIN	-----	PIN**	FNC **	LABEL
CLOCK	** CK	** 1-!		!-24	** +5V	**VCC
/BT	** I	** 2-!		!-23	** /B	**N/C
COMPARE	** I	** 3-!		!-22	** 0	**/START
/HLD	** I	** 4-!	P	!-21	** 0	**BIT11
/TRACK	** I	** 5-!	L	!-20	** 0	**BIT10
BIT4	** I	** 6-!	S	!-19	** 0	**BIT9
BIT3	** I	** 7-!	1	!-18	** 0	**BIT8
BIT2	** I	** 8-!	7	!-17	** 0	**BIT7
BIT1	** I	** 9-!	9	!-16	** 0	**BIT6
BIT0	** I	** 10-!		!-15	** 0	**BIT5
DONE	** I	** 11-!		!-14	** /B	**N/C
GND	** OV	** 12-!		!-13	** /OE	**N/C

TB033705

## High-Speed 12-Bit Tracking A/D Converter Using PLS179

AN28

## APPENDIX C: BOOLEAN EQUATIONS OF UP/DOWN COUNTER AND INPUT LATCH

```

File Name : ADCB1
Date : 10/21/1986
Time : 10:54:48

@DEVICE TYPE
PLS179
@DRAWING
@REVISION
@DATE
@SYMBOL
FILE NAME : ADCB1

@COMPANY
@NAME
@DESCRIPTION
@COMMON PRODUCT TERM
@COMPLEMENT ARRAY
@I/O DIRECTION
@OUTPUT POLARITY
@FLIP FLOP CONTROL
FC = 1 ; "SET ALL FLIP FLOP TO BE J/K"

@OUTPUT ENABLE
@REGISTER LOAD
@ASYNCHRONOUS PRESET/RESET
@FLIP FLOP MODE
@LOGIC EQUATION
"NON-INVERTING INPUT LATCH: /START = /ST "
START : J = ST ;
K = /ST ;
"UP/DOWN COUNTER ROUTINE"
/BITS : T = /START * TRACK * DONE * /HLD * COMPARE * /BIT0 * /BIT1 *
/BIT2 * /BIT3 * /BIT4 +
/START * TRACK * DONE * /HLD * /COMPARE * BIT0 * BIT1 *
BIT2 * BIT3 * BIT4 ;
/BIT6 : T = /START * TRACK * DONE * /HLD * COMPARE * /BIT0 * /BIT1 *
/BIT2 * /BIT3 * /BIT4 * /BITS +
/START * TRACK * DONE * /HLD * /COMPARE * BIT0 * BIT1 *
BIT2 * BIT3 * BIT4 * BITS ;
/BIT7 : T = /START * TRACK * DONE * /HLD * COMPARE *
/BIT0 * /BIT1 * /BIT2 * /BIT3 * /BIT4 * /BITS * /BIT6 +
/START * TRACK * DONE * /HLD * /COMPARE *
BIT0 * BIT1 * BIT2 * BIT3 * BIT4 * BITS * BIT6 ;
/BIT8 : T = /START * TRACK * DONE * /HLD * COMPARE * /BIT0 * /BIT1 *
/BIT2 * /BIT3 * /BIT4 * /BITS * /BIT6 * /BIT7 +
/START * TRACK * DONE * /HLD * /COMPARE * BIT0 * BIT1 *
BIT2 * BIT3 * BIT4 * BITS * BIT6 * BIT7 ;
/BIT9 : T = /START * TRACK * DONE * /HLD * COMPARE * /BIT0 * /BIT1 *
/BIT2 * /BIT3 * /BIT4 * /BITS * /BIT6 * /BIT7 * /BITS +
/START * TRACK * DONE * /HLD * /COMPARE * BIT0 * BIT1 *
BIT2 * BIT3 * BIT4 * BITS * BIT6 * BIT7 * BITS ;
/BIT10 : T = /START * TRACK * DONE * /HLD * COMPARE * /BIT0 * /BIT1 *
/BIT2 * /BIT3 * /BIT4 * /BITS * /BIT6 * /BIT7 * /BITS *
/BIT9 +
/START * TRACK * DONE * /HLD * /COMPARE * BIT0 * BIT1 *
BIT2 * BIT3 * BIT4 * BITS * BIT6 * BIT7 * BITS *
BIT9 ;
/BIT11 : T = /START * TRACK * DONE * /HLD * COMPARE * /BIT0 * /BIT1 *
/BIT2 * /BIT3 * /BIT4 * /BITS * /BIT6 * /BIT7 * /BITS *
/BIT9 *
/BIT10 +
/START * TRACK * DONE * /HLD * /COMPARE * BIT0 * BIT1 *
BIT2 * BIT3 * BIT4 * BITS * BIT6 * BIT7 * BITS *
BIT9 * BIT10 ;

```

TB039905

## High-Speed 12-Bit Tracking A/D Converter Using PLS179

AN28

## APPENDIX C: BOOLEAN EQUATIONS OF UP/DOWN COUNTER AND INPUT LATCH (Continued)

```

File Name : ADCB2
Date : 10/21/1986
Time : 10:58:56

@DEVICE TYPE
PLS179
@DRAWING
@REVISION
@DATE
@SYMBOL
FILE NAME : ADCB2

@COMPANY
@NAME
@DESCRIPTION
@COMMON PRODUCT TERM
@COMPLEMENT ARRAY
@I/O DIRECTION
DO = RC ; "RC OSCILLATOR"
D3 = DONE ; "ENABLE /DONE_OC TO OUTPUT A LOGIC LOW."
@OUTPUT POLARITY
@FLIP FLOP CONTROL
FC = 1 ;
@OUTPUT ENABLE
@REGISTER LOAD
@ASYNCHRONOUS PRESET/RESET
@FLIP FLOP MODE
"MO, M1, M2, M3, M4, M5 = 1 ; SET F0 - F5 TO J/K FLIP FLOPS."

@LOGIC EQUATION
"NON-INVERTING INPUT LATCH : /HLD = /HOLD "
HLD : J = HOLD ;
K = /HOLD ;

"UP/DOWN COUNTER ROUTINE"
/BIT0 : T = /START * TRACK * DONE * /HLD ;
/BIT1 : T = /START * TRACK * DONE * /HLD * /COMPARE * BIT0 +
/BIT2 : T = /START * TRACK * DONE * /HLD * /COMPARE * BIT0 * BIT1 +
/BIT3 : T = /START * TRACK * DONE * /HLD * /COMPARE * BIT0 * BIT1 *
BIT2 +
/BIT4 : T = /START * TRACK * DONE * /HLD * /COMPARE * BIT0 * BIT1 *
BIT2 *
BIT2 *
/BIT3 +
/BIT4 : T = /START * TRACK * DONE * /HLD * COMPARE * /BIT0 * /BIT1 * /
/BIT2 *
/DONE_OC = /( 1 ) ;

"RC OSCILLATOR"
RC = /(1) ;
CLOCK1 = RC ;
CLOCK2 = /( CLOCK1 ) ; "BUILT-IN DELAY OF 1 tPD"

```

TB039605



# High-Speed 12-Bit Tracking A/D Converter Using PLS179

AN28

## APPENDIX D: U1 ADCB1 FUSE MAP

```

File Name : ADCB1
Date : 10/21/1986
Time : 10:56:5

Cust/Project -
Date -
Rev/I. D. -

PLS179          ! F/F TYPE          ! E(b)= ! E(a) = ! POLARTY!
-----
T !          !          !          !          !          !          !
E !          !          !          !          !          !          !
R !          !          !          !          !          !          !
M ! C !          !          !          !          !          !          !
-----
0!A! 17_6_5_4_3_2_1_0!3_2_1_0!7_6_5_4_3_2_1_0!7_6_5_4_3_2_1_0!3_2_1_0!
1!A! 17_6_5_4_3_2_1_0!3_2_1_0!7_6_5_4_3_2_1_0!7_6_5_4_3_2_1_0!3_2_1_0!
2!A! L L L L, L H H ! - - H ! L ! - - - - - ! L ! - - - - - ! A A A A !
3!A! H H H H, L H L ! - - H ! L ! - - - - - ! L ! - - - - - ! A A A A !
4!A! L L L L, L H H ! - - H ! L ! - - - - - ! H ! - - - - - ! O ! A A A A !
5!A! H H H H, L H L ! - - H ! L ! - - - - - ! L ! - - - - - ! O ! A A A A !
6!A! L L L L, L H H ! - - H ! L ! - - - - - ! H H ! - - - - - ! O ! A A A A !
7!A! H H H H, L H L ! - - H ! L ! - - - - - ! L L ! - - - - - ! Q ! A A A A !
8!A! L L L L, L H H ! - - H ! L ! - - - - - ! H H H ! - - - - - ! 0 ! A A A A !
9!A! H H H H, L H L ! - - H ! L ! - - - - - ! L L L ! - - - - - ! 0 ! A A A A !
10!A! L L L L, L H H ! - - H ! L ! - - - - - ! H H H H ! - - - - - ! 0 ! A A A A !
11!A! H H H H, L H L ! - - H ! L ! - - - - - ! L L L L ! - - - - - ! 0 ! A A A A !
12!A! L L L L, L H H ! - - H ! L ! - - - - - ! H H H H H ! - - - - - ! 0 ! A A A A !
13!A! H H H H, L H L ! - - H ! L ! - - - - - ! L L L L L ! - - - - - ! 0 ! A A A A !
14!A! L L L L, L H H ! - - H ! L ! - - - - - ! H H H H H H ! - - - - - ! 0 ! A A A A !
15!A! H H H H, L H L ! - - H ! L ! - - - - - ! L L L L L ! - - - - - ! 0 ! A A A A !
16!A! 17_6_5_4_3_2_1_0!3_2_1_0!7_6_5_4_3_2_1_0!7_6_5_4_3_2_1_0!3_2_1_0!
17!A! L L L L, - H ! - - - - - ! L L L L H H H H H H H ! - - - - - ! A A A A !
18!A! L L L L, - L ! - - - - - ! L L L L H H H H H H H ! - - - - - ! A A A A !
19!A! L L L L, - H ! - - - - - ! L L L L L H H H H H H ! - - - - - ! A A A A !
20!A! L L L L, - L ! - - - - - ! L L L L L H H H H H H ! - - - - - ! A A A A !
21!A! L L L L, - H ! - - - - - ! L L L L - L H H H H H ! - - - - - ! H L - - - - - ! A A A A !
22!A! L L L L, - L ! - - - - - ! L L L L - L H H H H H ! - - - - - ! L - - - - - ! A A A A !
23!A! L L L L, - H ! - - - - - ! L L L L - - L H H H H ! - - - - - ! H L - - - - - ! A A A A !
24!A! L L L L, - L ! - - - - - ! L L L L - - L H H H H ! - - - - - ! L - - - - - ! A A A A !
25!A! L L L L, - H ! - - - - - ! L L L L - - L H H ! - - - - - ! H L - - - - - ! A A A A !
26!A! L L L L, - L ! - - - - - ! L L L L - - L H H ! - - - - - ! L H H ! - - - - - ! A A A A !
27!A! L L L L, - H ! - - - - - ! L L L L - - L H ! - - - - - ! H L ! A A A A !
28!A! L L L L, - L ! - - - - - ! L L L L - - L H ! - - - - - ! L H ! - - - - - ! A A A A !
29!A! L L L L, - H ! - - - - - ! L L L L - - L ! - - - - - ! H ! A A A A !
30!0!0 0 0 0,0 0 0!0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!
31!0!0 0 0 0,0 0 0!0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!
Fc!A! - - - - - ! - - - - - ! - - - - - ! - - - - - !
Pb! :!0 0 0 0,0 0 0!0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!
Rb! :!0 0 0 0,0 0 0!0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!
Lb! :!0 0 0 0,0 0 0!0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!
Pa! :!0 0 0 0,0 0 0!0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!
Ra! :!0 0 0 0,0 0 0!0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!
La! :!0 0 0 0,0 0 0!0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!
D3! :!0 0 0 0,0 0 0!0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!
D2! :!0 0 0 0,0 0 0!0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!
D1! :!0 0 0 0,0 0 0!0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!
D0! :!0 0 0 0,0 0 0!0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!0 0 0 0!

      BBBB / / C / N N D B / B B B B B B B B / B B B B B B B N N D B
      I I I I T H O S / / O I S T I I T I I I S I I T I I I I / 7 O I
      T T T T R L H T C C N T T T T T T T T T T T T T T T T T T C C N T
      1 2 3 4 A D P      E O A 1 1 9 8 7 6 5 A 1 1 9 8 7 6 5      E O
      C      A      R 1 0      R 1 0
      K      R      T      T
      E

TB029005
    
```

# High-Speed 12-Bit Tracking A/D Converter Using PLS179

## AN28

### APPENDIX D: U2 ADCB2 FUSE MAP (Continued)

```

File Name : ADCB2
Date : 10/21/1986
Time : 11:0:7

Cust./Project -
Date -
Rev./I. D. -

```

PLS179		F/F TYPE				E(b) = !E(a) = !POLARTY!						
T	:As:As:As:As:As:As:				0	0	!L:Ls:Hs:L!					
R	I	B(i)	Q(p)	Q(n)	B(o)							
M	C											
0!	7	6	5	4	3	2	1	0!	3	2	1	0!
1!A!	-	-	-	-	-	-	-	-	-	-	-	-
2!A!	-	-	-	-	-	-	-	-	-	-	-	-
3!A!	-	-	-	-	-	-	-	-	-	-	-	-
4!A!	-	-	-	-	-	-	-	-	-	-	-	-
5!A!	-	-	-	-	-	-	-	-	-	-	-	-
6!A!	-	-	-	-	-	-	-	-	-	-	-	-
7!A!	-	-	-	-	-	-	-	-	-	-	-	-
8!A!	-	-	-	-	-	-	-	-	-	-	-	-
9!A!	-	-	-	-	-	-	-	-	-	-	-	-
10!A!	-	-	-	-	-	-	-	-	-	-	-	-
11!A!	-	-	-	-	-	-	-	-	-	-	-	-
12!A!	-	-	-	-	-	-	-	-	-	-	-	-
13!A!	-	-	-	-	-	-	-	-	-	-	-	-
14!A!	-	-	-	-	-	-	-	-	-	-	-	-
15!A!	-	-	-	-	-	-	-	-	-	-	-	-
16!A!	-	-	-	-	-	-	-	-	-	-	-	-
17!A!	-	-	-	-	-	-	-	-	-	-	-	-
18!A!	-	-	-	-	-	-	-	-	-	-	-	-
19!A!	-	-	-	-	-	-	-	-	-	-	-	-
20!A!	-	-	-	-	-	-	-	-	-	-	-	-
21!A!	-	-	-	-	-	-	-	-	-	-	-	-
22!A!	-	-	-	-	-	-	-	-	-	-	-	-
23!A!	-	-	-	-	-	-	-	-	-	-	-	-
24!A!	-	-	-	-	-	-	-	-	-	-	-	-
25!A!	-	-	-	-	-	-	-	-	-	-	-	-
26!A!	-	-	-	-	-	-	-	-	-	-	-	-
27!0!	0	0	0	0	0	0	0	0	0	0	0	0
28!0!	0	0	0	0	0	0	0	0	0	0	0	0
29!0!	0	0	0	0	0	0	0	0	0	0	0	0
30!0!	0	0	0	0	0	0	0	0	0	0	0	0
31!0!	0	0	0	0	0	0	0	0	0	0	0	0
Fc!	A!	A!	A!	A!	A!	A!	A!	A!	A!	A!	A!	A!
Pb!	.	.	.	.	.	.	.	.	.	.	.	.
Rb!	.	.	.	.	.	.	.	.	.	.	.	.
Lb!	.	.	.	.	.	.	.	.	.	.	.	.
Pa!	.	.	.	.	.	.	.	.	.	.	.	.
Ra!	.	.	.	.	.	.	.	.	.	.	.	.
La!	.	.	.	.	.	.	.	.	.	.	.	.
D3!	.	.	.	.	.	.	.	.	.	.	.	.
D2!	.	.	.	.	.	.	.	.	.	.	.	.
D1!	.	.	.	.	.	.	.	.	.	.	.	.
D0!	.	.	.	.	.	.	.	.	.	.	.	.

```

NNNB //C //C CRN /BBBBBBDN /BBBBBBD /CCR
//ITHOSD L L C /H I I I I O /H I I I I O D L L C
CC C TRD T O D O C L T T T T T N C L T T T T T N O O O
      5 A L P A N C C D 4 3 2 1 0 E D 4 3 2 1 0 E N C C
      C D A R E K K
      K R T _ 2 1
      E O
      C

```

TR033905



## AN29 PLHS501 Programmable Macro Logic Primer

Application Note

### Application Specific Products

#### SUMMARY

The evolution of Programmable Logic Devices (PLD's) has led to the birth of a new generation of programmable devices designated as PML (Programmable Macro Logic). The immense versatility of these devices brings them closer as plausible alternatives to semicustom design approaches in low-to-medium ranges of applications. The following paper begins with a background on PML and a brief description of the PML basic architecture (See Reference 1.) Next, the first PML devices are introduced with a detailed discus-

sion of the PLHS501. The implementation of PML in the AMAZE software package is presented. A system level example intends to demonstrate the capabilities of PML as an eloquent and efficient design alternative.

#### THE EMERGENCE OF THE THIRD GENERATION PLD ARCHITECTURE

PML was introduced at WESCON '85 by Signetics Corporation. The unique architecture of PML breaks away into a new era of

programmable logic devices. The purpose of the PML architecture is to overcome the two level AND-OR bottleneck and provide the user with a higher level of logic integration. Current PLD's rely on two levels of logic transformation to implement combinational logic in Sum-Of-Products (SOP) form. In addition various PLD's make use of higher level macros such as flip-flops to form sequential logic functions. These macros connect the AND-OR chain to dedicated I/O pins.

Figure 1 show the basic architecture of one of the most recent PAL™ devices. It is clear that this architecture is inefficient in making full use of the available on-chip resources. This is due to the fact that an unused I/O macro will be wasted and remains futile.

For example, if an I/O pin is used as an input, the output macros are all wasted. Obviously, such an architecture cannot provide the user with an increase in the levels of logic integration. The PML device takes advantage of the fundamental architecture shown in Figure 2 to overcome these deficiencies and waste of on-chip resources. As shown in Figure 2, PML incorporates the NAND-NAND gate equivalence to break the AND-OR bottleneck.

The core of the PML is the programmable NAND-NAND network which connects the input and output macros to each other. Thus the inputs, outputs, and function macros are all connected by a single array.

At the present, two devices are under development which employ the above architecture. Both devices are fabricated by the Signetics state-of-the-art oxide isolation process and are packaged in 52-pin PLCC.

The first device available will be the PLHS501. The seemingly simple structure of this device can implement every logic function furnished by the current PAL™/PLA devices. Although the PLHS501 is principally a combinational logic device, its unique architecture makes it an ideal tool for applications involving asynchronous state machines (See Reference 2.)

The PLHS502 is a sequential logic device. It supplements the PLHS501 features with sixteen edge-triggered flip-flops. The device can provide diverse applications encompassing synchronous and asynchronous state machine designs.

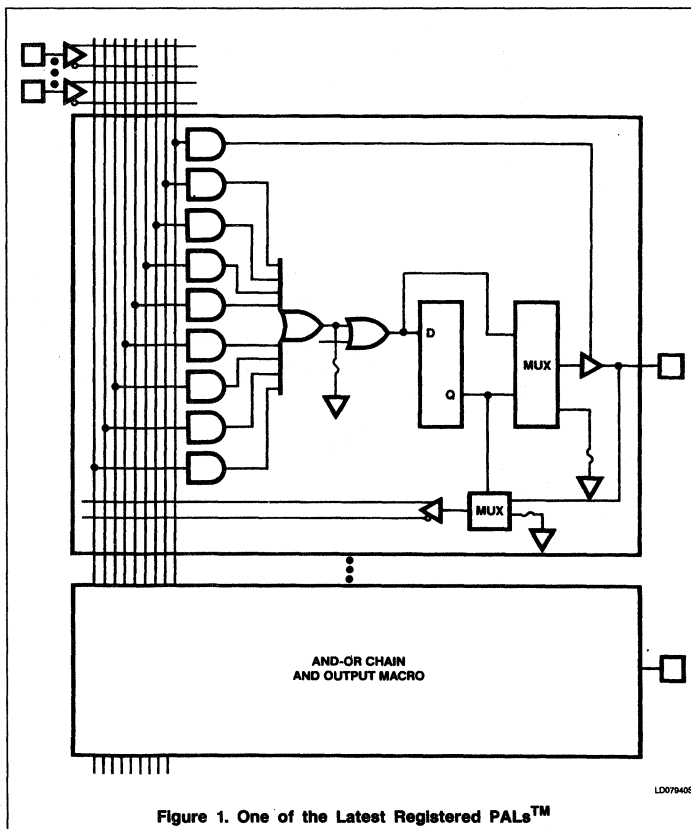


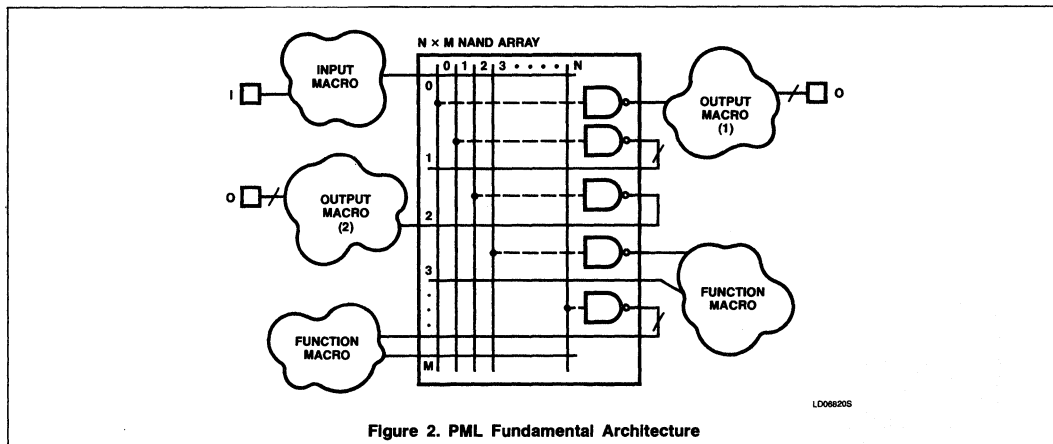
Figure 1. One of the Latest Registered PALs™

# PLHS501 Programmable Macro Logic Primer

AN29

**Table 1. Functional Description of PLHS501 I/O Pins**

PIN NO.	IDENTIFIER	FUNCTION
41-45, 47-52, 1-7, 9-14	I0-I23	Dedicated inputs
37-40	/B0-/B3	Fuse-programmable bidirectional I/Os with Active-Low outputs. Can be configured as open-collector outputs.
15-18	B4-B7	Logic controlled bidirectional I/Os with Active-High Tri-state outputs.
28-29 30-31 32-33 35-36	X0-X1 X2-X3 X4-X5 X6-X7	Pairs of Tri-state Exclusive-OR outputs that have common Output Enable.
19, 21 22-23	O0-O1 O2-O3	Pairs of dedicated Active-Low Tri-state output buffers. Each pair has common Output Enable control.
24-25 26-27	O4-O5 O6-O7	Pairs of dedicated Active-High Tri-state output buffers. Each pair has common Output Enable control.



# PLHS501 Programmable Macro Logic Primer

AN29

## THE PLHS501 PML

The PLHS501 architecture in Figure 3 exhibits an exquisite logic tool. The device provides a combination of 72 NAND terms, 24 dedicated inputs (I0-I23), eight bidirectional I/O's (B0-B7), eight exclusive-OR outputs (X0-X7),

and eight dedicated outputs (O0-O7). Figure 4 shows the PLHS501 logic diagram and Table 1 illustrates the functional breakdown of the PLHS501 I/O pins.

Since the output of each NAND term feeds back to the inputs of the NAND array, intri-

cate logic functions can be implemented without wasting valuable I/O pins. For example, in order to implement an internal 'RS' latch in a combinational PAL™/PLD, at least two inputs and two outputs are required. The same internal latch can be configured by the PLHS501 without using any I/O pins.

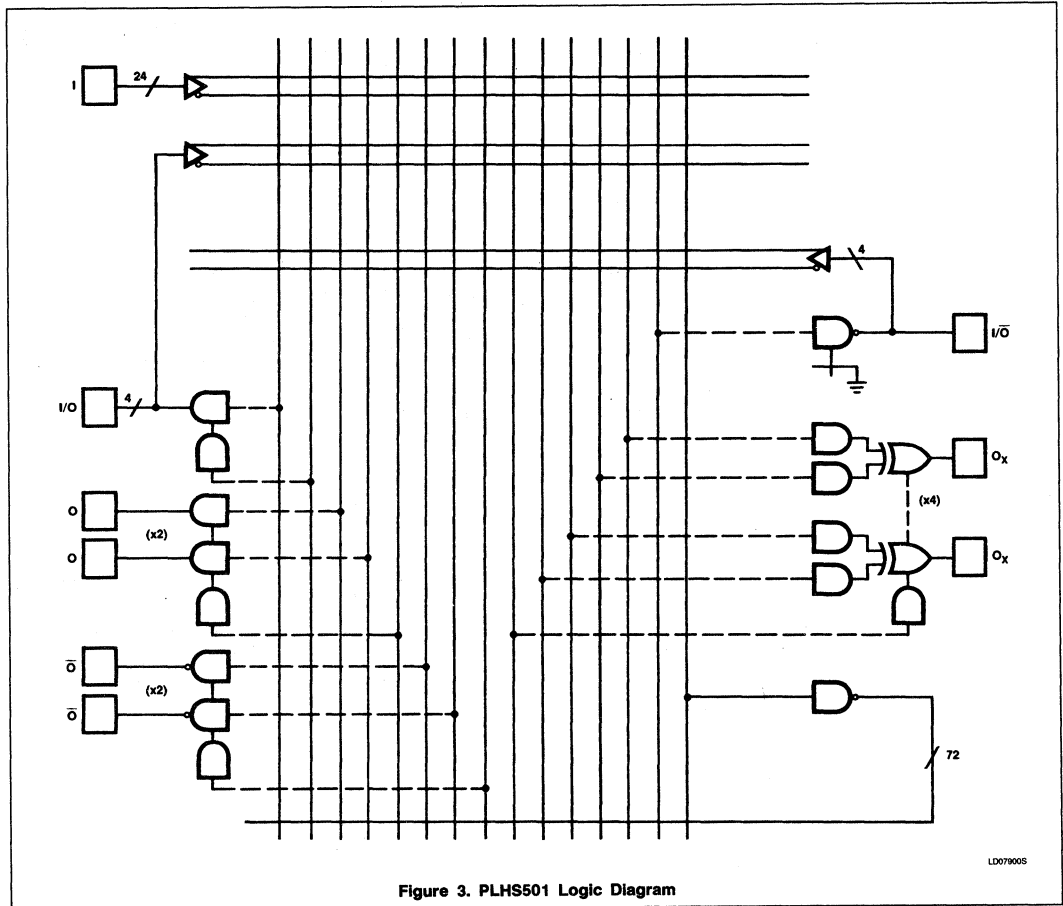


Figure 3. PLHS501 Logic Diagram

LD079005

# PLHS501 Programmable Macro Logic Primer

# AN29

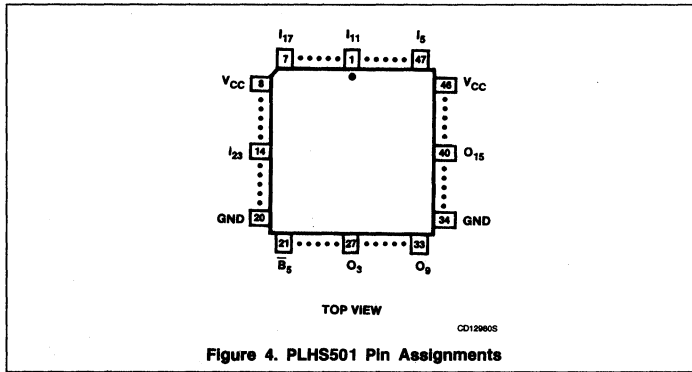


Figure 4. PLHS501 Pin Assignments

Figure 5 illustrates how 'RS' and 'D' latches are implemented in the PLHS501.

Another eminent application of the PLHS501 is in generating asynchronous state machines.

The blend of internal feedback paths together with the abundant number of gates makes this device suitable for designing asynchronous state machines which employ propagation delays of feedback paths as memory elements as shown in Figure 6. (See Reference 2.)

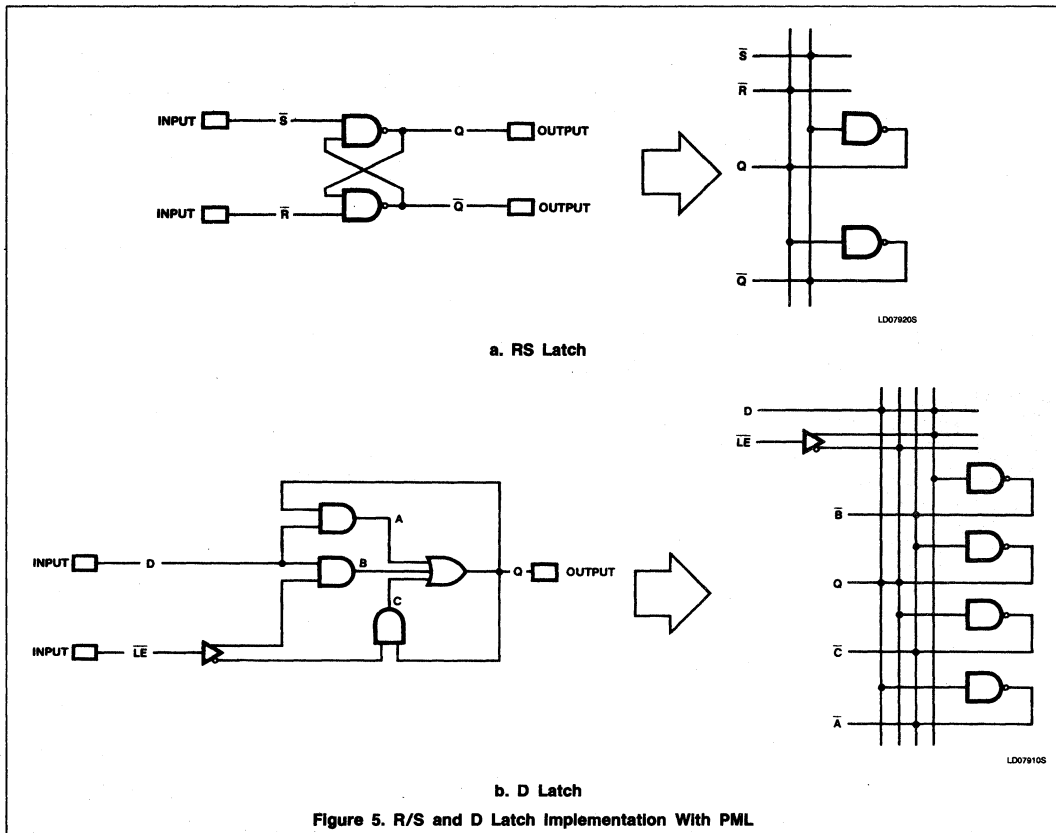


Figure 5. R/S and D Latch Implementation With PML

# PLHS501 Programmable Macro Logic Primer

AN29

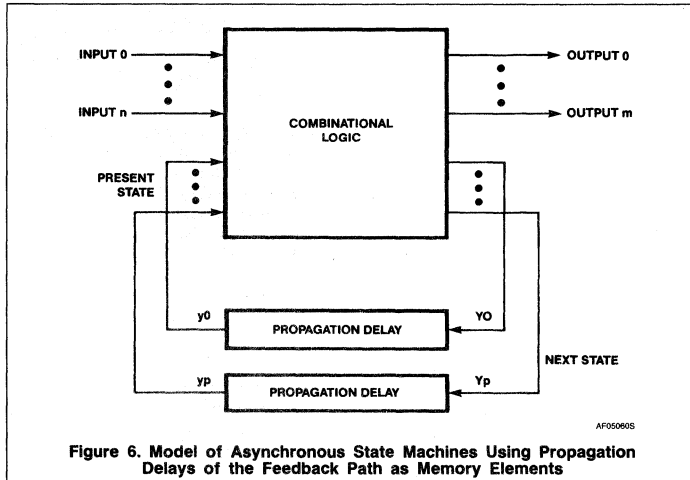


Figure 6. Model of Asynchronous State Machines Using Propagation Delays of the Feedback Path as Memory Elements

## PML DEVELOPMENT SOFTWARE

Programmable logic development software has become an integral part of the PLD design process. Without software tools PLDs become perplexing devices which are inconvenient to use. Development software enables the user to take full advantage of the programmable logic's resources. The complexity of the PML devices makes software an indispensable element in the design process.

The AMAZE PLD design software, as noted in Reference 3, has been developed for Signetics programmable logic devices. PML

design and development will be fully supported by AMAZE. Figure 7 shows the AMAZE configuration for supporting PML. The structure of the software is based on the following modules:

- .BLAST (Boolean Logic And State Transfer entry)
- .DPI (Device Programmer Interface)
- .SIM (PLD functional SIMulator)
- .PI (PML Integrator)

The foundation for the above modules is an AMAZE standard fuse file (STD). The STD file is the common means of communication between all the modules. The following para-

graphs briefly explain the implementation of PML in the AMAZE structure.

### BLAST

The basic elements of BLAST are:

1. Boolean Equation Entry (BEE).
2. State Equation Entry (SEE).
3. Schematic To Boolean Converter (STBC).

1. BEE accepts user defined logic in the form of boolean equations and produces an AMAZE standard fuse file. The boolean notation in BEE will account for the following PML features.

a. The single array architecture provides an option to specify the number of logic (gate) levels that accommodate a specified function. The number of gate levels dictate the delay between the inputs and outputs. The number of logic levels for each function can be specified using the following notation:

LABEL [ number of logic levels ] = equation; where LABEL is the user designated name of the output pin.

b. Some outputs have more than one gate associated with them (for example registered or EXOR outputs). The input SIDE of these outputs can be specifically defined. For example:

$$\begin{aligned} \text{Pin Label: } R &= \text{eqn}; \\ S &= \text{eqn}; \end{aligned}$$

c. User definable CLOCK Logic (e.g. as in the PLHS502) can be specified.

d. For PML devices the Sum-of-Products (SOP) can be defined as common terms.

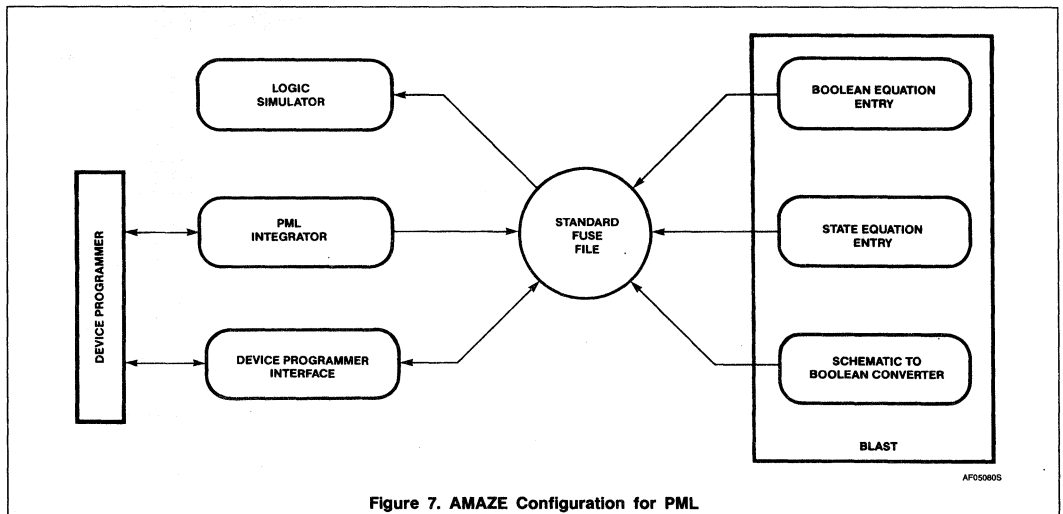


Figure 7. AMAZE Configuration for PML

## PLHS501 Programmable Macro Logic Primer

AN29

2. SEE accepts state machine definitions and produces the AMAZE standard fuse file. Particularly, SEE will include asynchronous state machine implementations utilizing the PLHS501.

3. STBC convert schematic netlists produced by CAD systems such as Futurenet™, Daisy, and Mentor into an AMAZE BEE file. STBC will fully embody the PML in the following manner:

a. De Morgan's theorem is applied to produce equations in SOP form. All the functions in SOP form are then converted into their NAND- NAND equivalent.

b. The converter will determine the polarity and the number of levels in each equation and will automatically determine the correct output polarity.

### DPI

Device Programmer Interface provides the interface between the AMAZE standard fuse file and a commercial programmer. It allows the transmission of data to and from the device programmer.

### SIM

The functional simulator uses the AMAZE standard fuse file in the following manner:

a. An event driven simulator will assess the delays within the PML in order to properly simulate the pattern.

b. Automatically generates test vectors for the pattern simulation.

### PI

The PML Integrator is a conversion that transforms various PAL™ or PLD circuits into a PML device. It will automatically fit multiple PAL™/PLD devices into a single PML. It is capable of automatically receiving patterns from a commercial device programmer and downloading it back to the programmer after the PML transformation. The implementation of an 'Integrator' will allow the automatic conversion of numerous PAL™/PLDs into a single PML device.

AMAZE will not be the total extent of development software available for the PML devices. The task of implementing PML design software is already underway by a number of different vendors of CAD software.

### PLHS501 DESIGN EXAMPLE

The following example intends to manifest the capabilities of the PLHS501. Figure 8 shows a system formed with TTL logic. The system requirements make it imperative only to use discrete asynchronous latches. Thus, none of the 7 latches in the system can be directly replaced by registers. The system diagram is drafted using Futurenet™ DASH-2 Schematic Designer. The system is partitioned into two PLS173s and one PLS153. In order to convert the system into its targeted PLD's, the PINLIST (see Figures 9a-9c) has to be defined. Using the AMAZE PIN-LIST editor, the specified PLD's are labeled with the same labels as those on the system schematic (Figure 8). After the declaration of labels, AMAZE automatically converts the system to the designated PLDs. The AMAZE

generated boolean equation files are shown in Figures 9a, 9b, and 9c. Figure 10 shows the overall system implemented with PLDs. The logic condensation capabilities of PML makes it feasible to replace the whole system by a single PLHS501 (Figure 11). The PLHS501 in this design will still have ample space for any future additions.

The above example demonstrates only part of the PLHS501 capabilities. The introduction of PML devices and their immense logic power will pave the way for a new generation of efficient and elegant systems.

### REFERENCES

1. Cavlan, Napoleone 1985. "Third Generation PLD Architecture Breaks AND-OR Bottleneck", WESCON 1985 Conference Proceedings.
2. Wong, David K. "Third Generation PLS Architecture and its Applications", Electro 1986 Conference Proceedings.
3. "1986 PLD Data Manual", Signetics Corporation.



# PLH501 Programmable Macro Logic Primer

## AN29

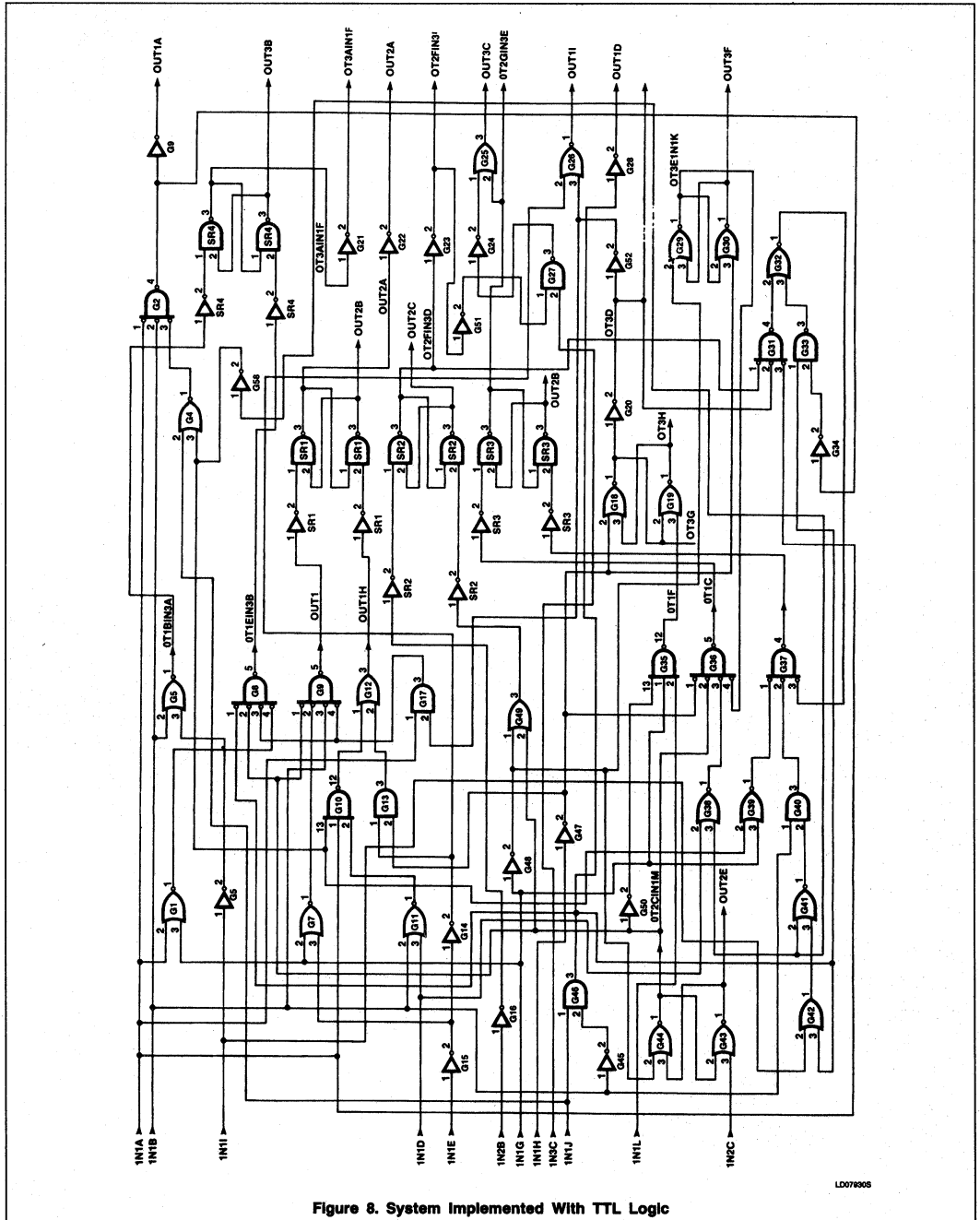


Figure 8. System Implemented With TTL Logic

LD079305

## PLHS501 Programmable Macro Logic Primer

AN29

File Name : PART1  
Date : 12/10/1986  
Time : 18:25:54

## \*\*\*\*\* P I N L I S T \*\*\*\*\*

LABEL	** FNC	**PIN	-----	PIN**	FNC **	LABEL
IN1A	** I	** 1-		-24	** +5V	**VCC
IN1B	** I	** 2-		-23	** 0	**OUT1I
OT3G	** I	** 3-		-22	** 0	**OUT1H
IN1D	** I	** 4-	P	-21	** 0	**OUT1G
IN1E	** I	** 5-	L	-20	** 0	**OT1F
OT3AIN1F	** I	** 6-	S	-19	** 0	**OT1EIN3B
IN1G	** I	** 7-	1	-18	** 0	**OUT1D
IN1H	** I	** 8-	7	-17	** 0	**OT1C
IN1I	** I	** 9-	3	-16	** 0	**OT1BIN3A
IN1J	** I	** 10-		-15	** 0	**OUT1A
OT3EIN1K	** I	** 11-		-14	** I	**OT2CIN1M
GND	** OV	** 12-		-13	** I	**IN1L

TB03405

File Name : PART1  
Date : 12/10/1986  
Time : 18:26:56

## @DEVICE TYPE

PLS173

## @DRAWING

## @REVISION

## @DATE

## @SYMBOL

## @COMPANY

## @NAME

PART1

## @DESCRIPTION

## @I/O DIRECTION

## @LOGIC EQUATION

```

OUT1A = /IN1A*/IN1B*(IN1J+/OT3AIN1F);
OT1BIN3A = /IN1B*IN1I;
OT1C = /IN1H+OT2CIN1M+/IN1D*/OT3AIN1F+OT3EIN1K;
OUT1D = IN1B+/IN1J;
OT1EIN3B = /IN1B*IN1J+OT2CIN1M+/OT3G*IN1A+/IN1A*/IN1G;
OT1F = OT2CIN1M*/IN1G*/IN1L;
OUT1G = OT2CIN1M+/IN1G*IN1E+IN1B+/OT3G*IN1A;
OUT1H = /IN1H*(IN1E)+OT3AIN1F+/IN1A+IN1B+IN1D;
OUT1I = (/IN1E)*OT3G;

```

TB034105

Figure 9a. Part 1: PLS173



PLHS501 Programmable Macro Logic Primer

AN29

File Name : PART2  
 Date : 12/10/1986  
 Time : 18:30:42

\*\*\*\*\* P I N L I S T \*\*\*\*\*

LABEL	** FNC	**PIN	-----	PIN** FNC	** LABEL
IN1A	** I	** 1-		-24 ** +5V	**VCC
IN2B	** I	** 2-		-23 ** B	**N/C
IN2C	** I	** 3-		-22 ** B	**N/C
IN1G	** I	** 4-	P	-21 ** 0	**OT2GIN3E
IN1I	** I	** 5-	L	-20 ** 0	**/OT2FIN3D
IN1J	** I	** 6-	S	-19 ** 0	**OT2CIN1M
IN1B	** I	** 7-		-18 ** 0	**OUT2E
OUT1G	** I	** 8-		-17 ** 0	**OUT2D
OUT1H	** I	** 9-		-16 ** 0	**OUT2C
OT1C	** I	** 10-		-15 ** 0	**OUT2B
OT3AIN1F	** I	** 11-		-14 ** /0	**/OUT2A
GND	** OV	** 12-		-13 ** I	**OT3G

TB03420S

File Name : PART2  
 Date : 12/10/1986  
 Time : 18:31:49

@DEVICE TYPE  
 PLS173  
 @DRAWING  
 @REVISION  
 @DATE  
 @SYMBOL  
 @COMPANY  
 @NAME  
 PART2

@DESCRIPTION  
 @I/O DIRECTION  
 @LOGIC EQUATION

```

/OUT2A = /((OUT2B*/OUT1G));
OUT2B =OUT1H+OUT2A;
OUT2C =OT2CIN1M+/IN1G+OT2FIN3D;
OUT2D =OT3AIN1F*/IN1G+(/OT3AIN1F*((IN1B+/IN1J)*/IN1I))*IN1B
      +(OT2FIN3D*/OT3G*/IN1A)*((/IN1A*/IN1B*(IN1J+/OT3AIN1F))
      *(/IN1B*IN1J))+/OT2GIN3E;
OUT2E =/OT2CIN1M*/IN2C;
OT2CIN1M =IN1G*/OUT2E;
/OT2FIN3D = /((OUT2C*(IN2B)));
OT2GIN3E =/OUT2D+OT1C;
    
```

TB03430S

Figure 9b. Part 2: PLS173

PLHS501 Programmable Macro Logic Primer

AN29

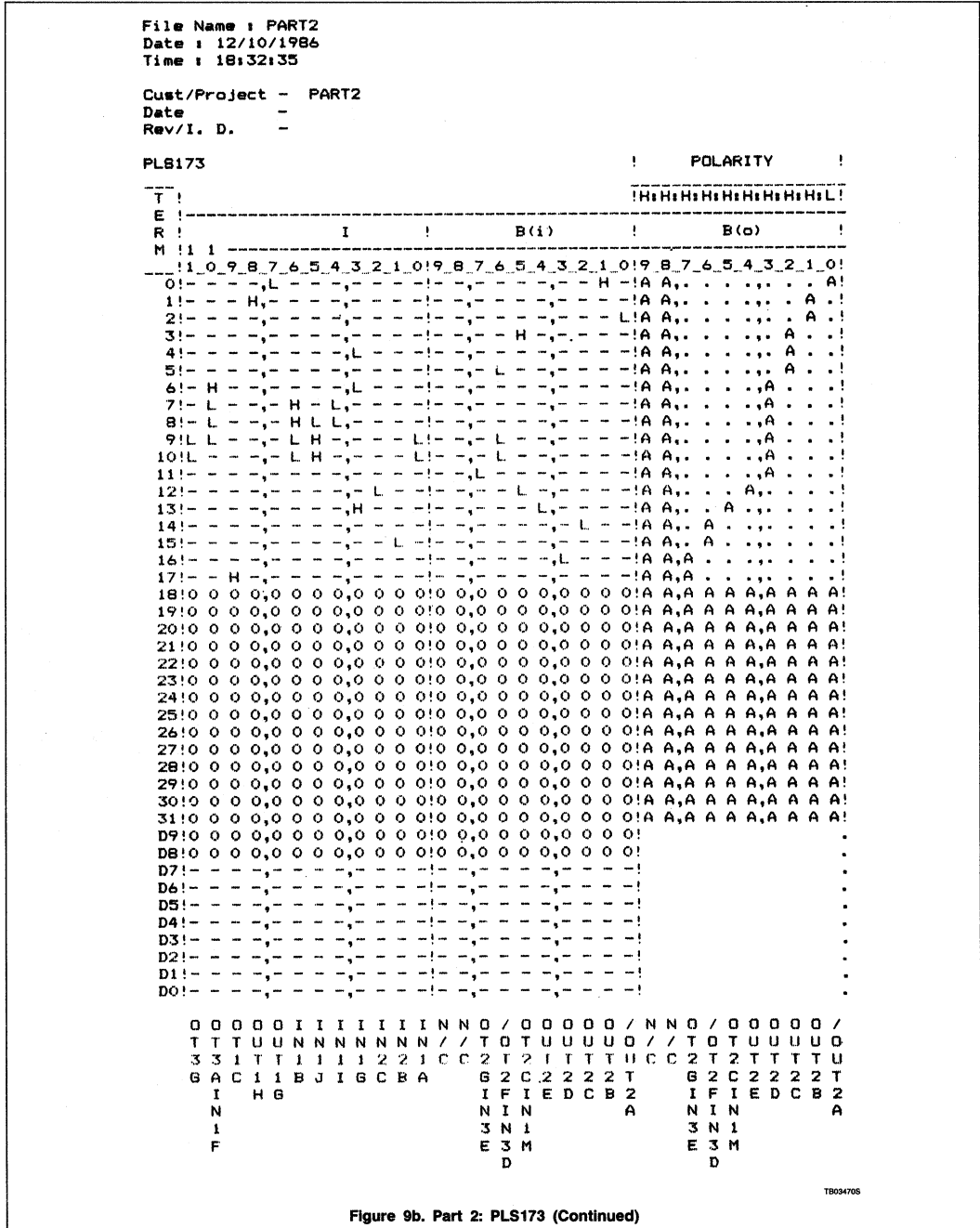


Figure 9b. Part 2: PLS173 (Continued)

PLHS501 Programmable Macro Logic Primer

AN29

File Name : PART3  
 Date : 12/10/1986  
 Time : 18:35:38

\*\*\*\*\* P I N L I S T \*\*\*\*\*

LABEL	** FNC	**PIN	-----	PIN**	FNC **	LABEL
OT1BIN3A	** I	** 1-		1-20	** +5V	**VCC
OT1EIN3B	** I	** 2-		1-19	** B	**N/C
IN3C	** I	** 3-	P	1-18	** B	**N/C
OT2IN3D	** I	** 4-	L	1-17	** 0	**OT3H
OT2GIN3E	** I	** 5-	S	1-16	** /0	**/OT3G
OT1F	** I	** 6-	1	1-15	** 0	**OUT3F
IN1L	** I	** 7-	5	1-14	** 0	**OT3EIN1K
IN1H	** I	** 8-	3	1-13	** 0	**OUT3D
/OT3AIN1F	** /0	** 9-		1-12	** 0	**OUT3C
GND	** 0V	** 10-		1-11	** I	**IN1G

TB03405

File Name : PART3  
 Date : 12/10/1986  
 Time : 18:36:18

- @DEVICE TYPE
- PLS153
- @DRAWING
- @REVISION
- @DATE
- @SYMBOL
- @COMPANY
- @NAME
- @DESCRIPTION
- @COMMON PRODUCT TERM
- @I/O DIRECTION
- @OUTPUT POLARITY
- @LOGIC EQUATION

TB03405

Figure 9c. Part 3: PLS153

# PLHS501 Programmable Macro Logic Primer

# AN29

```

File Name : PART3
Date : 12/10/1986
Time : 18:36:47

Cust/Project -
Date -
Rev/I. D. -

PLS153                                     ! POLARITY !
-----
T !                                     !HrHrHrLrLrHrHrHrLrLr!
E !
-----
R !           !           B(1)           !           B(0)           !
M !
-----
 7_6_5_4_3_2_1_0!9_8_7_6_5_4_3_2_1_0!9_8_7_6_5_4_3_2_1_0!
0!0 0 0 0,0 0 0!0 0,0 0 0,0 0 0 0!A A,A A A,A A A A A!
1!0 0 0 0,0 0 0!0 0,0 0 0,0 0 0 0!A A,A A A,A A A A A!
2!0 0 0 0,0 0 0!0 0,0 0 0,0 0 0 0!A A,A A A,A A A A A!
3!0 0 0 0,0 0 0!0 0,0 0 0,0 0 0 0!A A,A A A,A A A A A!
4!0 0 0 0,0 0 0!0 0,0 0 0,0 0 0 0!A A,A A A,A A A A A!
5!0 0 0 0,0 0 0!0 0,0 0 0,0 0 0 0!A A,A A A,A A A A A!
6!0 0 0 0,0 0 0!0 0,0 0 0,0 0 0 0!A A,A A A,A A A A A!
7!0 0 0 0,0 0 0!0 0,0 0 0,0 0 0 0!A A,A A A,A A A A A!
8!0 0 0 0,0 0 0!0 0,0 0 0,0 0 0 0!A A,A A A,A A A A A!
9!0 0 0 0,0 0 0!0 0,0 0 0,0 0 0 0!A A,A A A,A A A A A!
10!0 0 0 0,0 0 0!0 0,0 0 0,0 0 0 0!A A,A A A,A A A A A!
11!0 0 0 0,0 0 0!0 0,0 0 0,0 0 0 0!A A,A A A,A A A A A!
12!0 0 0 0,0 0 0!0 0,0 0 0,0 0 0 0!A A,A A A,A A A A A!
13!0 0 0 0,0 0 0!0 0,0 0 0,0 0 0 0!A A,A A A,A A A A A!
14!0 0 0 0,0 0 0!0 0,0 0 0,0 0 0 0!A A,A A A,A A A A A!
15!0 0 0 0,0 0 0!0 0,0 0 0,0 0 0 0!A A,A A A,A A A A A!
16!0 0 0 0,0 0 0!0 0,0 0 0,0 0 0 0!A A,A A A,A A A A A!
17!0 0 0 0,0 0 0!0 0,0 0 0,0 0 0 0!A A,A A A,A A A A A!
18!0 0 0 0,0 0 0!0 0,0 0 0,0 0 0 0!A A,A A A,A A A A A!
19!0 0 0 0,0 0 0!0 0,0 0 0,0 0 0 0!A A,A A A,A A A A A!
20!0 0 0 0,0 0 0!0 0,0 0 0,0 0 0 0!A A,A A A,A A A A A!
21!0 0 0 0,0 0 0!0 0,0 0 0,0 0 0 0!A A,A A A,A A A A A!
22!0 0 0 0,0 0 0!0 0,0 0 0,0 0 0 0!A A,A A A,A A A A A!
23!0 0 0 0,0 0 0!0 0,0 0 0,0 0 0 0!A A,A A A,A A A A A!
24!0 0 0 0,0 0 0!0 0,0 0 0,0 0 0 0!A A,A A A,A A A A A!
25!0 0 0 0,0 0 0!0 0,0 0 0,0 0 0 0!A A,A A A,A A A A A!
26!0 0 0 0,0 0 0!0 0,0 0 0,0 0 0 0!A A,A A A,A A A A A!
27!0 0 0 0,0 0 0!0 0,0 0 0,0 0 0 0!A A,A A A,A A A A A!
28!0 0 0 0,0 0 0!0 0,0 0 0,0 0 0 0!A A,A A A,A A A A A!
29!0 0 0 0,0 0 0!0 0,0 0 0,0 0 0 0!A A,A A A,A A A A A!
30!0 0 0 0,0 0 0!0 0,0 0 0,0 0 0 0!A A,A A A,A A A A A!
31!0 0 0 0,0 0 0!0 0,0 0 0,0 0 0 0!A A,A A A,A A A A A!
D9!0 0 0 0,0 0 0!0 0,0 0 0,0 0 0 0!
D8!0 0 0 0,0 0 0!0 0,0 0 0,0 0 0 0!
D7!- - - -!- - -!- - -!- - -!
D6!- - - -!- - -!- - -!- - -!
D5!- - - -!- - -!- - -!- - -!
D4!- - - -!- - -!- - -!- - -!
D3!- - - -!- - -!- - -!- - -!
D2!- - - -!- - -!- - -!- - -!
D1!0 0 0 0,0 0 0!0 0,0 0 0,0 0 0 0!
D0!- - - -!- - -!- - -!- - -!

I I 0 0 0 I 0 0 N N 0 / 0 0 0 0 I / N N 0 / 0 0 0 0 I /
N N T T T N T T / / T O U T U U N D / / T O U T U U N D /
J J 1 2 2 3 1 J C C 3 T T 3 T T 1 T C C 3 T T 3 T T 1 T
H L F G I C E B H 3 3 E 3 3 G 3 H 3 3 E 3 3 G 3
      I N I I      G F I D C A      G F I D C A
      N 3 N N      N      I      N      I
      3 D 3 3      1      N      1      N
      E B A      K      1      K      1
                        F

```

T8034005

Figure 9c. Part 3: PLS153 (Continued)

PLHS501 Programmable Macro Logic Primer

AN29

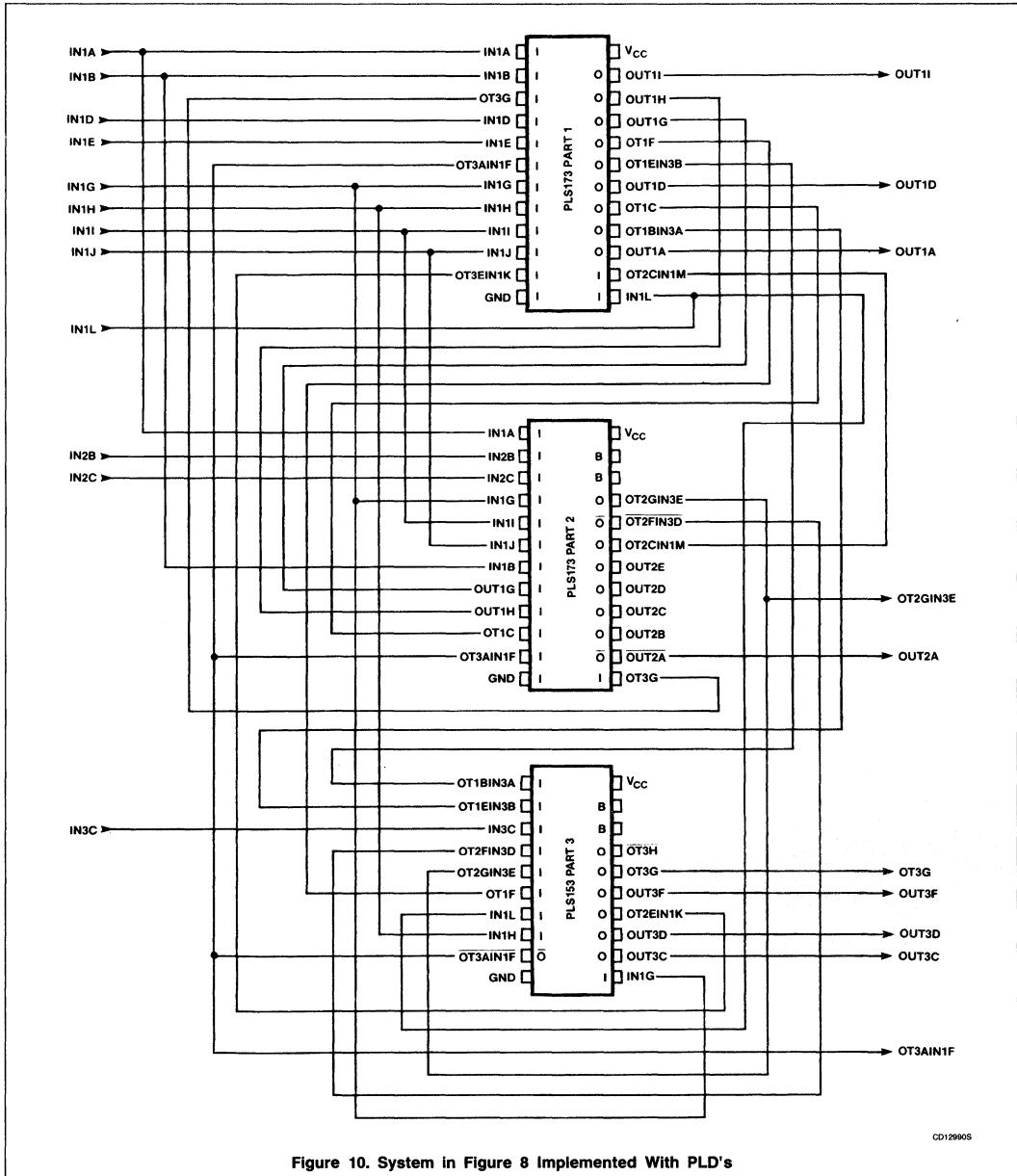


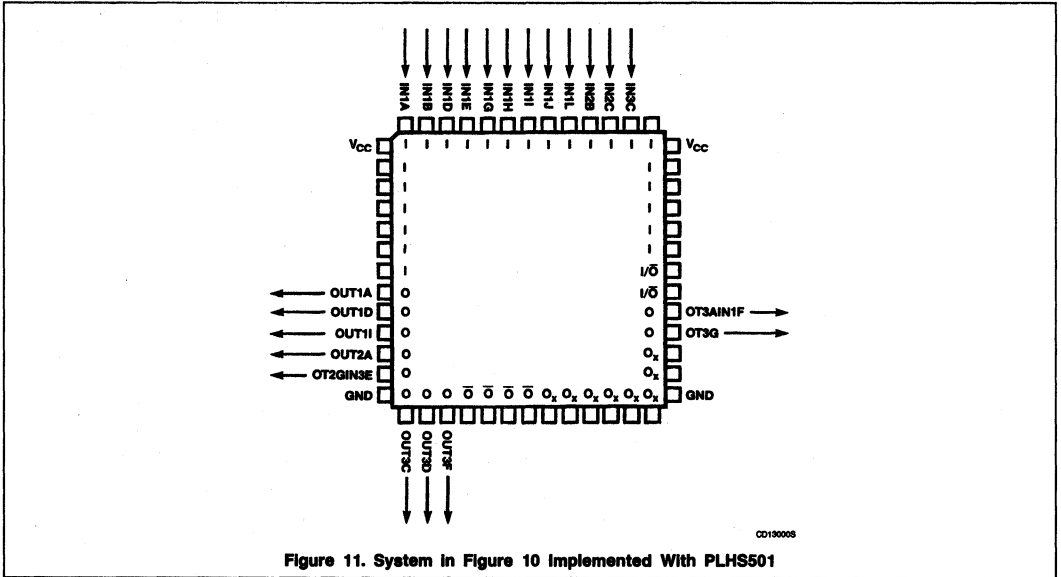
Figure 10. System in Figure 8 Implemented With PLD's

CD129905



# PLHS501 Programmable Macro Logic Primer

AN29



### INDEX

#### Section 10

##### Package Outlines

A	PLASTIC: Leaded Chip Carrier .....	10-3
F	CERAMIC: Dual-In-Line .....	10-5
FA	CERAMIC: Dual-In-Line With Quartz Window .....	10-7
N	PLASTIC: Dual-In-Line .....	10-9



# Package Outlines

## PLASTIC PLCC

1. Package dimensions conform to JEDEC specifications for standard Leaded Chip Carrier outline (PLCC) package.
2. Controlling dimensions are given in inches with dimensions in millimeters contained in parentheses.
3. Dimensions and tolerancing per ANSI Y14.5M - 1982.
4. "D-E" and "F-G" are reference datums on the molded body and do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15mm (0.006") on any side.
5. Pin numbers start with pin #1 and continue counterclockwise when viewed from the top.
6. Lead material: Olin 194 (Copper Alloy) or equivalent, solder dipped.
7. Body material: Plastic (Epoxy).
8. Thermal resistance values are determined by temperature sensitive parameter (TSP) method. This method uses the

forward voltage drop of a calibrated diode to measure the change in junction temperature due to a known power application. Test condition for these values follow:

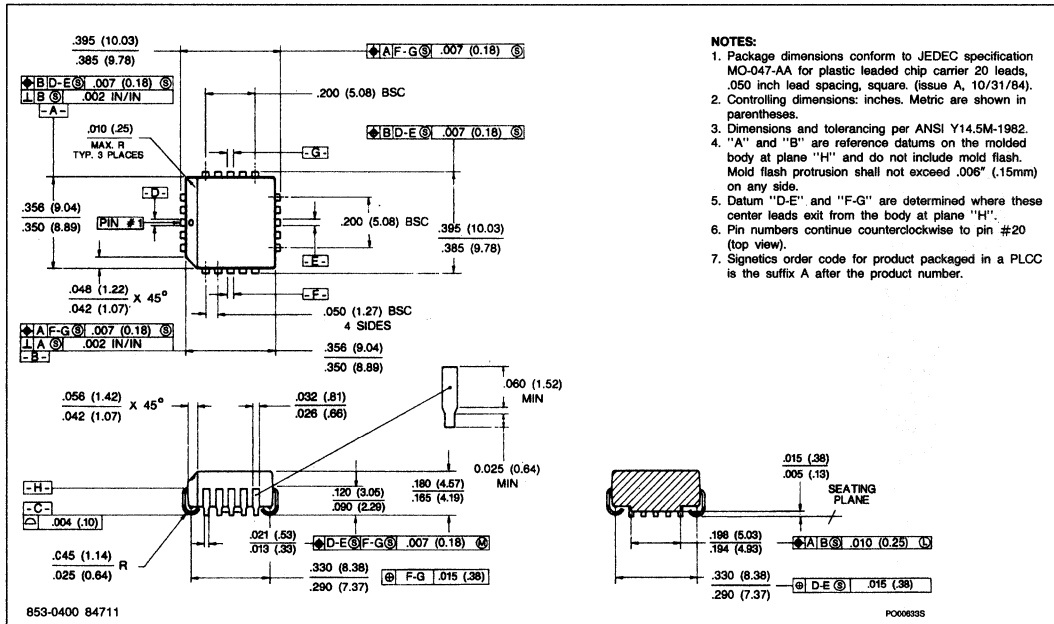
Test Ambient — Still Air  
 Test Fixture —  $\theta_{JA}$  - Glass epoxy test board (2.24" × 2.24" × 0.062")  
 $\theta_{JC}$  - Water cooled heat sink

## PLASTIC LEADED CHIP CARRIER (PLCC)

NO. OF LEADS	PACKAGE CODE	DESCRIPTION
20	A	350mil-wide
28	A	450mil-wide
52	A	750mil-wide

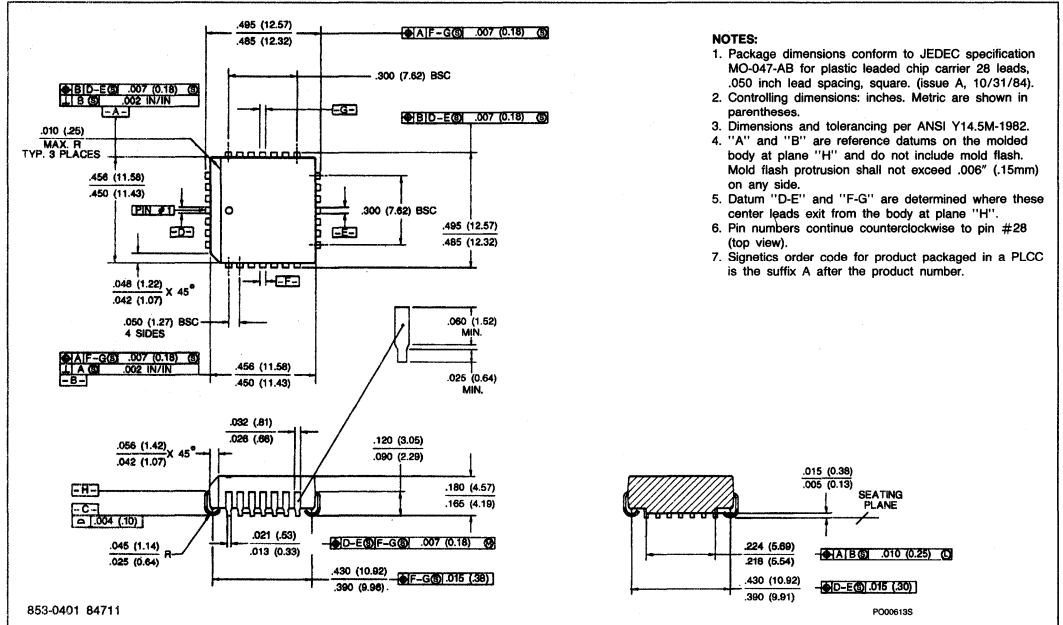
TYPICAL $\theta_{JA}/\theta_{JC}$ VALUES (°C/W)			
Die Size	Power Dissipation (W)	Average $\theta_{JA}$	Average $\theta_{JC}$
20K	.75	70	30
25K	1.0	61	23
50K	1.0	42	15

## 20-PIN PLASTIC PLCC (A PACKAGE)

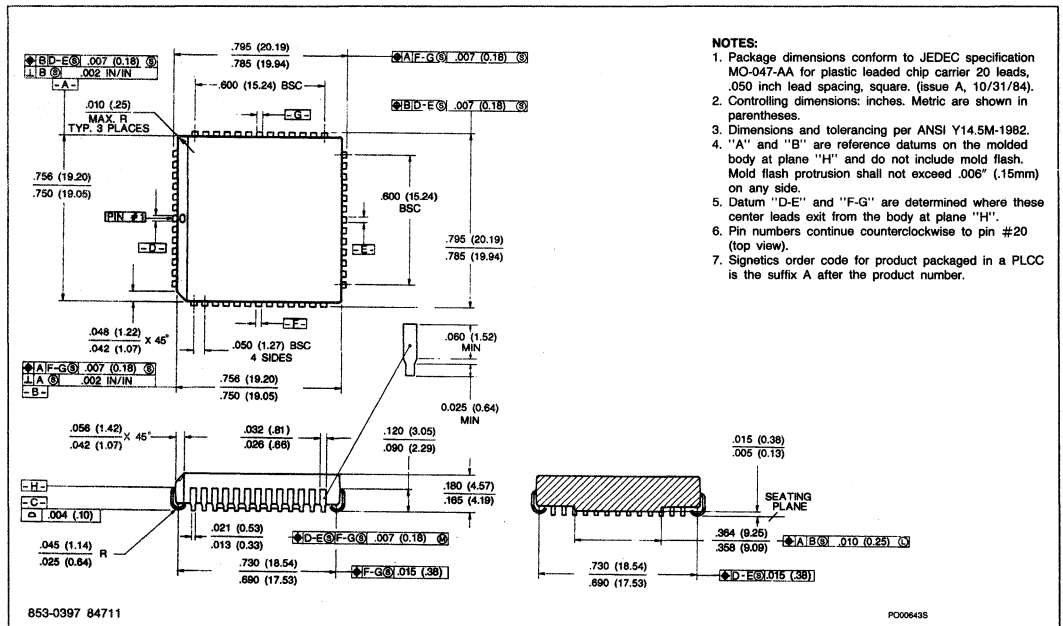


# Package Outlines

## 28-PIN PLASTIC PLCC (A PACKAGE)



## 52-PIN PLASTIC PLCC (A PACKAGE)



## Package Outlines

### HERMETIC CERDIP

1. Package dimensions conform to JEDEC specifications for standard Ceramic Dual Inline (Cerdip) package.
2. Controlling dimensions are given in inches with dimensions in millimeters, mm, contained in parentheses.
3. Dimensions and tolerancing per ANSI Y14.5M - 1982.
4. Pin numbers start with pin #1 and continue counterclockwise when viewed from the top.
5. These dimensions measured with the leads constrained to be perpendicular to plane T.
6. Lead material: ASTM alloy F-30 (Alloy 42) or equivalent - tin plated or solder dipped.
7. Body Material: Ceramic with glass seal at leads.
8. Thermal resistance values are determined by temperature sensitive parameter (TSP) method. This method uses the

forward voltage drop of a calibrated diode to measure the change in junction temperature due to a known power application. Test condition for these values follow:

Test Ambient — Still Air

Test Fixture —  $\theta_{JA}$  - Textool ZIF socket with 0.04" stand-off

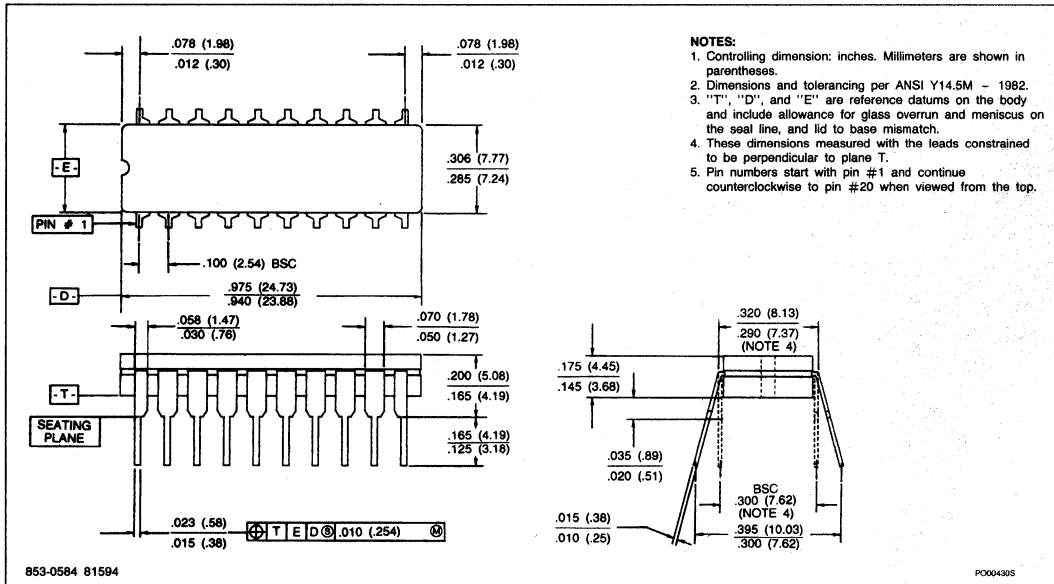
$\theta_{JC}$  - Water cooled heat sink

### HERMETIC DUAL-IN-LINE PACKAGES

NO. OF LEADS	PACKAGE CODE	DESCRIPTION
20	F	300mil-wide
24	F	300mil-wide
28	F	600mil-wide

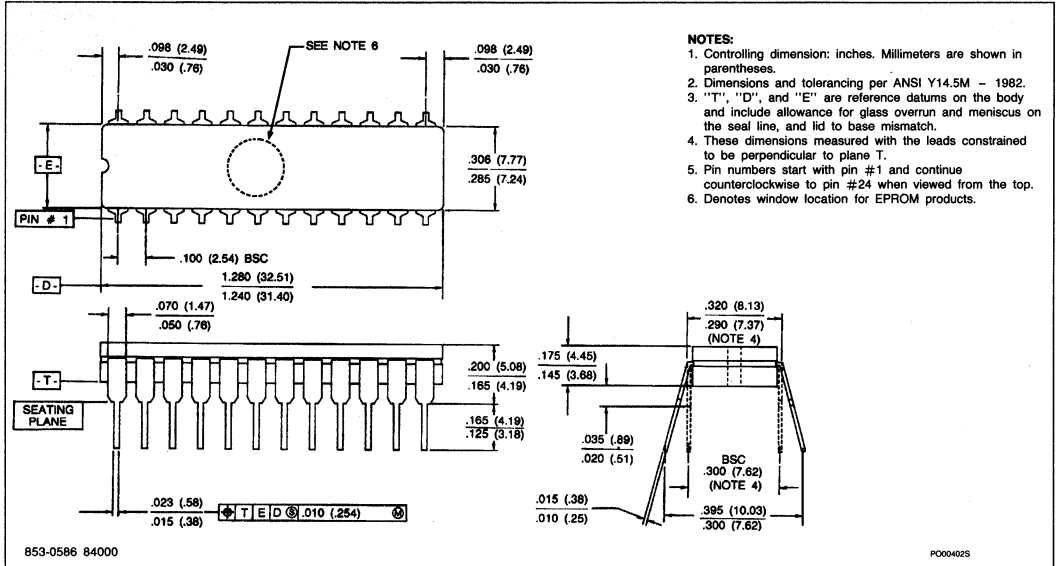
TYPICAL $\theta_{JA}/\theta_{JC}$ VALUES (°C/W)			
Die Size	Power Dissipation (W)	Average $\theta_{JA}$	Average $\theta_{JC}$
20K	.75	70	7.8
25K	1.0	62	6.9
30K	1.0	48	6.3

### 20-PIN CERAMIC DIP (F PACKAGE)

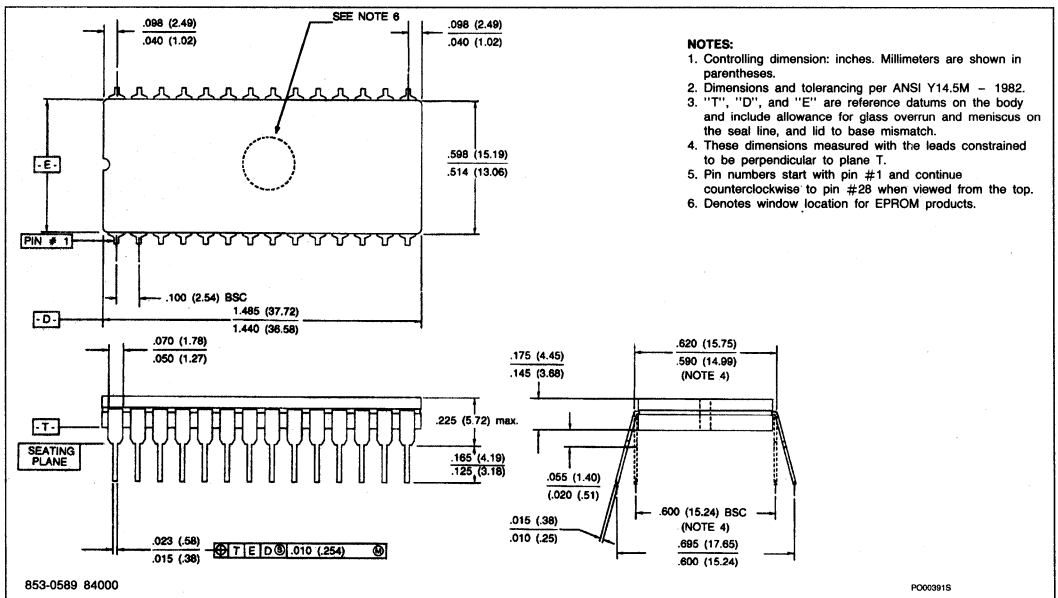


## Package Outlines

### 24-PIN CERAMIC DIP (F PACKAGE)



### 28-PIN CERAMIC DIP (F PACKAGE)



## Package Outlines

### HERMETIC CERDIP WITH QUARTZ WINDOW

- Package dimensions conform to JEDEC specifications for standard Ceramic Dual Inline (Cerdip) package.
- Controlling dimensions are given in inches with dimensions in millimeters, mm, contained in parentheses.
- Dimensions and tolerancing per ANSI Y14.5M - 1982.
- Pin numbers start with pin #1 and continue counterclockwise when viewed from the top.
- These dimensions measured with the leads constrained to be perpendicular to plane T.
- Lead material: ASTM alloy F-30 (Alloy 42) or equivalent - tin plated or solder dipped.
- Body Material: Ceramic with glass seal at leads.
- Thermal resistance values are determined by temperature sensitive parameter (TSP) method. This method uses the

forward voltage drop of a calibrated diode to measure the change in junction temperature due to a known power application. Test condition for these values follow:

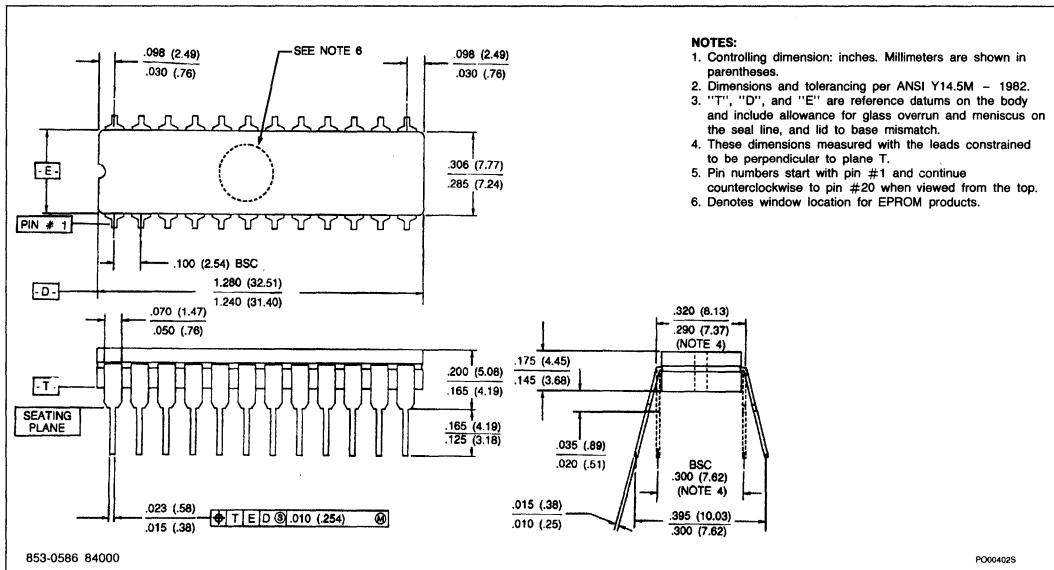
- Test Ambient — Still Air
- Test Fixture —  $\theta_{JA}$  - Textool ZIF socket with 0.04" standoff
- $\theta_{JC}$  - Water cooled heat sink

### HERMETIC DUAL-IN-LINE PACKAGES WITH QUARTZ WINDOW

NO. OF LEADS	PACKAGE CODE	DESCRIPTION
24	F	300mil-wide
28	F	600mil-wide

TYPICAL $\theta_{JA}/\theta_{JC}$ VALUES ( $^{\circ}\text{C}/\text{W}$ )			
Die Size	Power Dissipation (W)	Average $\theta_{JA}$	Average $\theta_{JC}$
25K	.5	67	7.8
30K	.5	52	7.0

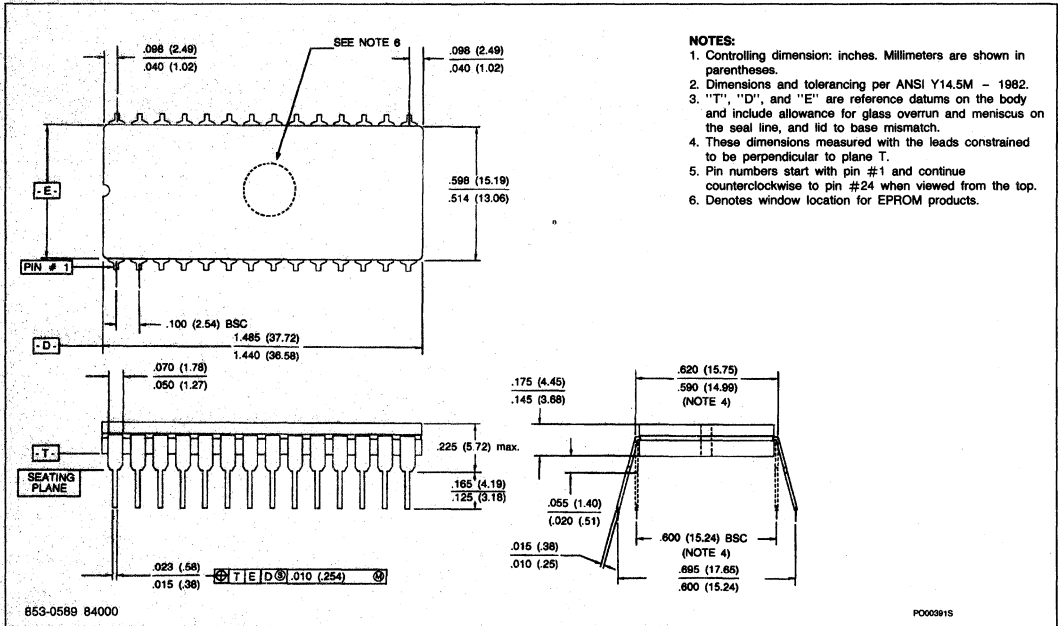
### 24-PIN CERAMIC DIP WITH QUARTZ WINDOW (FA PACKAGE)





# Package Outlines

## 28-PIN CERAMIC DIP WITH QUARTZ WINDOW (FA PACKAGE)



## Package Outlines

### PLASTIC DIP

- Package dimensions conform to JEDEC specification MS-001-AA for standard Plastic Dual Inline (DIP) package.
- Controlling dimensions are given in inches with dimensions in millimeters, mm, contained in parentheses.
- Dimensions and tolerancing per ANSI Y14.5M - 1982.
- "T", "D" and "E" are reference datums on the molded body and do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.01 inch (0.25mm) on any side.
- These dimensions measured with the leads constrained to be perpendicular to plane T.
- Pin numbers start with pin #1 and continue counterclockwise when viewed from the top.
- Lead material: Olin 194 (Copper Alloy) or equivalent, solder dipped.
- Body material: Plastic (Epoxy)
- Thermal resistance values are determined by temperature sensitive parameter (TSP) method. This method uses the

forward voltage drop of a calibrated diode to measure the change in junction temperature due to a known power application. Test condition for these values follow:

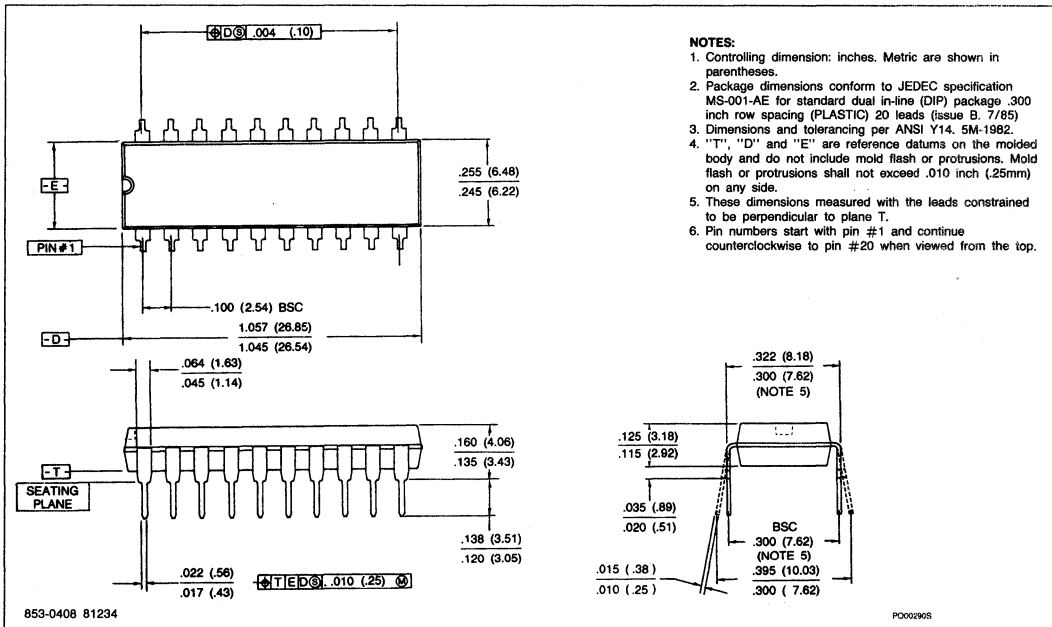
Test Ambient — Still Air  
 Test Fixture —  $\theta_{JA}$  - Textool ZIF socket with 0.04" stand-off  
 $\theta_{JC}$  - Water cooled heat sink

### PLASTIC DUAL-IN-LINE PACKAGES

NO. OF LEADS	PACKAGE CODE	DESCRIPTION
20	N	Cu. Lead Frame 300mil-wide
24	N	Cu. Lead Frame 300mil-wide
28	N	Cu. Lead Frame 600mil-wide

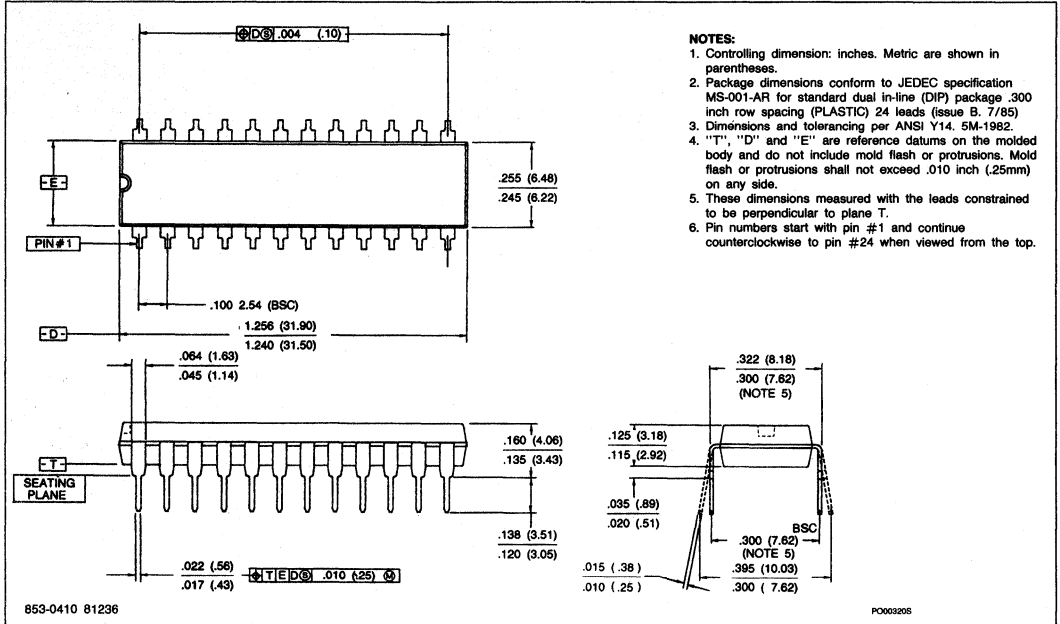
TYPICAL $\theta_{JA}/\theta_{JC}$ VALUES ( $^{\circ}\text{C}/\text{W}$ )			
Die Size	Power Dissipation (W)	Average $\theta_{JA}$	Average $\theta_{JC}$
20K	.75	61	25
25K	1.0	55	25
50K	1.0	46	18

### 20-PIN PLASTIC (N PACKAGE)

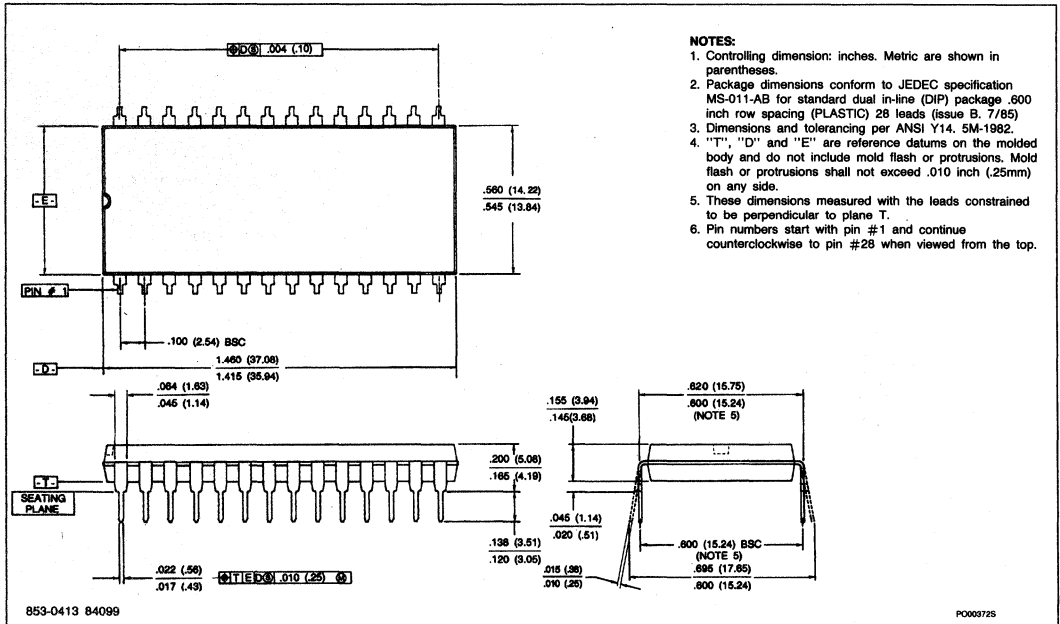


# Package Outlines

## 24-PIN PLASTIC (N PACKAGE)



## 28-PIN PLASTIC (N PACKAGE)



# Electronic components and materials for professional, industrial and consumer uses from the world-wide Philips Group of Companies

**Argentina:** PHILIPS ARGENTINA S.A., Div. Elcoma, Vedia 3892, 1430 BUENOS AIRES, Tel. (01) 541 - 7141 to 7747.  
**Australia:** PHILIPS INDUSTRIES LTD., Elcoma Division, 11 Waltham Street, ARTARMON, N.S.W. 2064, Tel. (02) 439 3322.  
**Austria:** ÖSTERREICHISCHE PHILIPS INDUSTRIE G.m.b.H., UB Bauelemente, Triester Str. 64, A-1101 WIEN, Tel. (0222) 62 91 11-0.  
**Belgium:** N.V. PHILIPS & MBL ASSOCIATED, 80 Rue Des Deux Gares, B-1070 BRUXELLES, Tel. (02) 525-61-11.  
**Brazil:** CONSTANTA-IBRAPE; (Active Devices): Av. Brigadeiro Faria Lima, 1735-SAO PAULO-SP, Tel. (011) 211-2600,  
(Passive Devices & Materials): Av. Francisco Monteiro, 702 - RIBEIRAO PIRES-SP, Tel. (011) 459-8211.  
**Canada:** PHILIPS ELECTRONICS LTD., Elcoma Division, 601 Milner Ave., SCARBOROUGH, Ontario, M1B 1M8, Tel. (416) 292-5161.  
**Chile:** PHILIPS CHILENA S.A., Av. Santa Maria 0760, SANTIAGO, Tel. (02) 77 38 16.  
**Colombia:** IND. PHILIPS DE COLOMBIA S.A., c/o IPRELENCO LTD., Cra. 21, No. 56-17, BOGOTA, D.E., Tel. (01) 249 76 24.  
**Denmark:** MINIWATT A/S, Strandlodsvej 2, P.O. Box 1919, DK 2300 COPENHAGEN S, Tel. (01) 54 11 33.  
**Finland:** OY PHILIPS AB, Elcoma Division, Kaivokatu 8, SF-00100 HELSINKI 10, Tel. (90) 1 72 71.  
**France:** RTC-COMPELEC, 130 Avenue Ledru Rollin, F-75540 PARIS 11, Tel. (01) 43 38 80 00.  
**Germany (Fed. Republic):** VALVO, UB Bauelemente der Philips G.m.b.H., Valvo Haus, Burchardstrasse 19, D-2 HAMBURG 1, Tel. (040) 3296-0.  
**Greece:** PHILIPS HELLENIQUE S.A., Elcoma Division, No. 15, 25th March Street, GR 17778 TAVROS, Tel. (01) 48 94 339/48 94 911.  
**Hong Kong:** PHILIPS HONG KONG LTD., Elcoma Div., 15/F Philips Ind. Bldg., 24-28 Kung Yip St., KWAI CHUNG, Tel. (0)-24 51 21.  
**India:** PEICO ELECTRONICS & ELECTRICALS LTD., Elcoma Dept., Band Box Building,  
254-D Dr. Annie Besant Rd., BOMBAY - 400 025, Tel. (022) 4930311/4930590.  
**Indonesia:** PT. PHILIPS-RALIN ELECTRONICS, Elcoma Div., Setiabudi II Building, 6th Fl., Jalan H.R. Rasuna Said (P.O. Box 223/KBY) Kuningan,  
JAKARTA 12910, Tel. (021) 51 79 95.  
**Ireland:** PHILIPS ELECTRICAL (IRELAND) LTD., Elcoma Division, Newstead, Clonskeagh, DUBLIN 14, Tel. (01) 69 33 55.  
**Italy:** PHILIPS S.p.A., Div. Componenti Elcoma, Piazza IV Novembre 3, I-20124 MILANO, Tel. (02) 6752.1.  
**Japan:** NIHON PHILIPS CORP., Shuwa Shinagawa Bldg., 26-33 Takanawa 3-chome, Minato-ku, TOKYO (108), Tel. (03) 448-5611.  
(IC Products) SIGNETICS JAPAN LTD., 8-7 Sanbancho Chiyoda-ku, TOKYO 102, Tel. (03) 230-1521.  
**Korea (Republic of):** PHILIPS ELECTRONICS (KOREA) LTD., Elcoma Div., Philips House, 260-199 Itaewon-dong, Yongsan-ku, SEOUL,  
Tel. (02) 794-5011.  
**Malaysia:** PHILIPS MALAYSIA SDN BHD, Elcoma Div., 345 Jalan Gelugor, 11700 PULAU PINANG, Tel. (04) 87 00 44.  
**Mexico:** ELECTRONICA, S.A. de C.V., Carr. México-Toluca km. 62.5, TOLUCA, Edo. de México 50140, Tel. Toluca 91 (721) 613-00.  
**Netherlands:** PHILIPS NEDERLAND, Marktgroep Elonco, Postbus 90050, 5600 PB EINDHOVEN, Tel. (040) 78 37 49.  
**New Zealand:** PHILIPS NEW ZEALAND LTD., Elcoma Division, 110 Mt. Eden Road, C.P.O. Box 1041, AUCKLAND, Tel. (09) 605-914.  
**Norway:** NORSK A/S PHILIPS, Electronica Dept., Sandstuveien 70, OSLO 6, Tel. (02) 68 02 00.  
**Pakistan:** PHILIPS ELECTRICAL CO. OF PAKISTAN LTD., Philips Markaz, M.A. Jinnah Rd., KARACHI-3, Tel. (021) 72 57 72.  
**Peru:** CADESA, Av. Alfonso Ugarte 1268, LIMA 5, Tel. (014) 326070.  
**Philippines:** PHILIPS INDUSTRIAL DEV. INC., 2246 Pasong Tamo, P.O. Box 911, Makati Comm. Centre, MAKATI-RIZAL 3116, Tel. (02) 86 89 51 to 59.  
**Portugal:** PHILIPS PORTUGUESA S.A.R.L., Av. Eng. Duarte Pacheco 6, 1009 LISBOA Codex, Tel. (019) 68 31 21.  
**Singapore:** PHILIPS PROJECT DEV. (Singapore) PTE LTD., Elcoma Div., Lorong 1, Toa Payoh, SINGAPORE 1231, Tel. 35 02 000.  
**South Africa:** S.A. PHILIPS (Pty) LTD., EDAC Div., 3rd Floor Rainer House, Upper Railway Rd. & Ove St., New Doornfontein, JOHANNESBURG 2001,  
Tel. (011) 402-4600/07.  
**Spain:** MINIWATT S.A., Balmes 22, BARCELONA 7, Tel. (03) 301 63 12.  
**Sweden:** PHILIPS KOMPONENTER A.B., Lidingövägen 50, S-11584 STOCKHOLM 27, Tel. (08) 7821000.  
**Switzerland:** PHILIPS A.G., Elcoma Dept., Allmendstrasse 140-142, CH-8027 ZÜRICH, Tel. (01) 488 22 11.  
**Taiwan:** PHILIPS TAIWAN LTD., 150 Tun Hua North Road, P.O. Box 22978, TAIPEI, Taiwan, Tel. (02) 7120500.  
**Thailand:** PHILIPS ELECTRICAL CO. OF THAILAND LTD., 283 Silom Road, P.O. Box 961, BANGKOK, Tel. (02) 233-6330-9.  
**Turkey:** TÜRK PHILIPS TICARET A.S., Elcoma Department, İnönü Cad., No. 78-80, 80090 Ayazpasa ISTANBUL, Tel. (01) 143 59 10.  
**United Kingdom:** MULLARD LTD., Mullard House, Torrington Place, LONDON WC1E 7HD, Tel. (01) 580 6633.  
**United States:** (Active Devices & Materials) AMPEREX SALES CORP., Providence Pike, SLATERSVILLE, R.I. 02876, Tel. (401) 762-9000.  
(Passive & Electromech. Dev.) MEPCO/CENTRALAB, INC., 2001 West Blue Heron Blvd, RIVIERA BEACH, Florida 33404,  
Tel. (305) 881-3200.  
(IC Products) SIGNETICS CORPORATION, 811 East Arques Avenue, SUNNYVALE, CA 94088-3409, Tel. (408) 991-2000.  
**Uruguay:** LUZILECTRON S.A., Avda Uruguay 1287, P.O. Box 907, MONTEVIDEO, Tel. (02) 98 53 95.  
**Venezuela:** IND. VENEZOLANAS PHILIPS S.A., c/o MAGNETICA S.A., Calle 6, Ed. Las Tres Jotas, App. Post. 78117, CARACAS, Tel. (02) 239 39 31.

**For all other countries apply to:** Philips Electronic Components and Materials Division, International Business Relations, P.O. Box 218,  
5600 MD EINDHOVEN, The Netherlands, Telex 350000 phtcnl

AB54

© Philips Export B.V. 1987

This information is furnished for guidance, and with no guarantee as to its accuracy or completeness; its publication conveys no licence under any patent or other right, nor does the publisher assume liability for any consequence of its use; specifications and availability of goods mentioned in it are subject to change without notice; it is not to be reproduced in any way, in whole or in part, without the written consent of the publisher.

Printed in The Netherlands

Date of release: 2-87

9398 147 0001