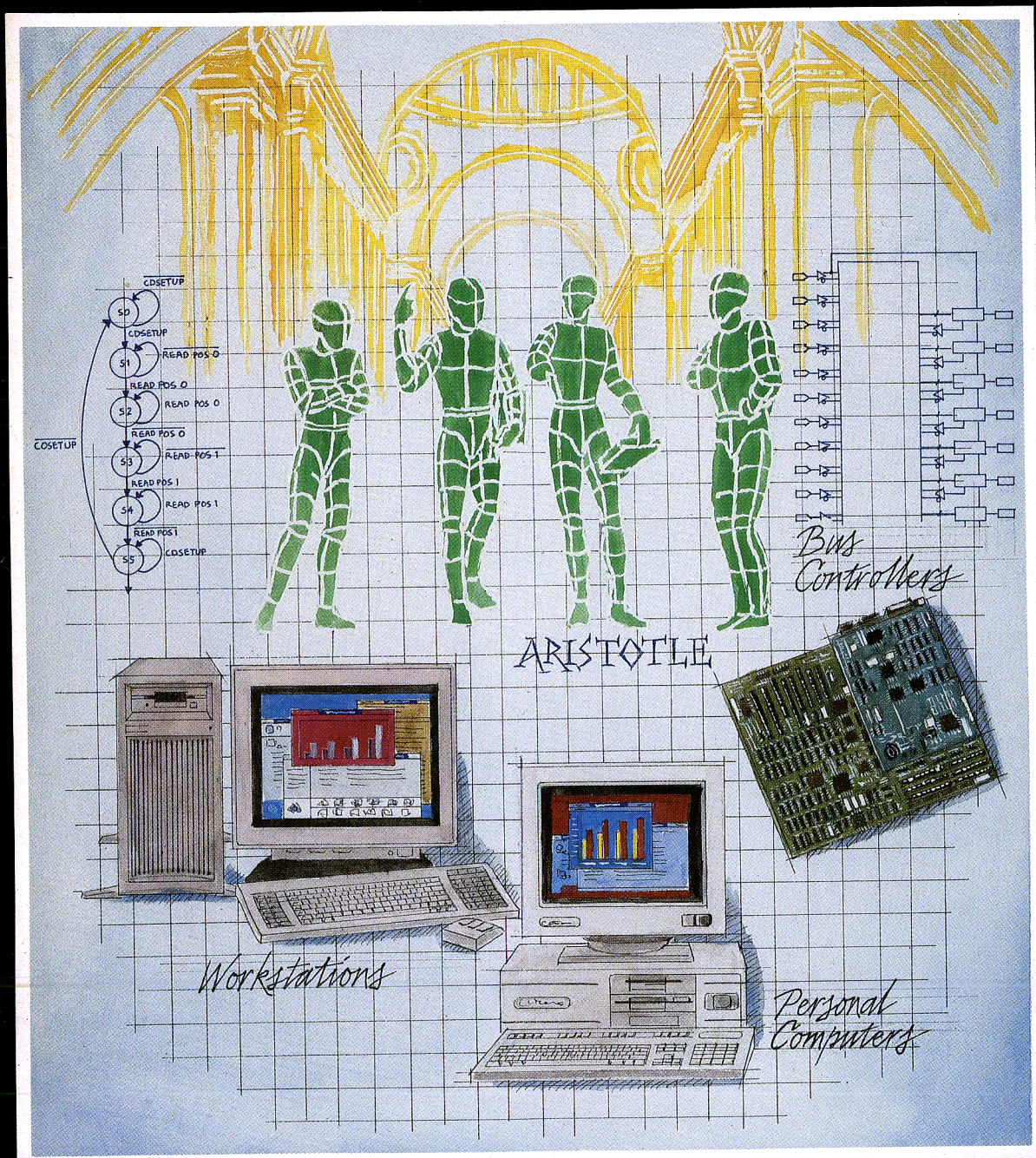
intel®

Programmable Logic



Benedict Norbert Wong

Order Number: 296083-006

LITERATURE

To order Intel Literature or obtain literature pricing information in the U.S. and Canada call or write Intel Literature Sales. In Europe and other international locations, please contact your ***local*** sales office or distributor.

INTEL LITERATURE SALES
P.O. BOX 7641
Mt. Prospect, IL 60056-7641

In the U.S. and Canada
call toll free
(800) 548-4725

CURRENT HANDBOOKS

Product line handbooks contain data sheets, application notes, article reprints and other design information. All handbooks can be ordered individually, and most are available in a pre-packaged set in the U.S. and Canada.

TITLE	INTEL ORDER NUMBER	ISBN
SET OF THIRTEEN HANDBOOKS (Available in U.S. and Canada)	**231003**	**N/A**
CONTENTS LISTED BELOW FOR INDIVIDUAL ORDERING:		
COMPONENTS QUALITY/RELIABILITY	210997	1-55512-132-2
EMBEDDED APPLICATIONS	270648	1-55512-123-3
8-BIT EMBEDDED CONTROLLERS	270645	1-55512-121-7
16-BIT EMBEDDED CONTROLLERS	270646	1-55512-120-9
16/32-BIT EMBEDDED PROCESSORS	270647	1-55512-122-5
MEMORY PRODUCTS	210830	1-55512-117-9
MICROCOMMUNICATIONS	231658	1-55512-119-5
MICROCOMPUTER PRODUCTS	280407	1-55512-118-7
MICROPROCESSORS	230843	1-55512-115-2
PACKAGING	240800	1-55512-128-4
PERIPHERAL COMPONENTS	296467	1-55512-127-6
PRODUCT GUIDE (Overview of Intel's complete product lines)	210846	1-55512-116-0
PROGRAMMABLE LOGIC	296083	1-55512-124-1
ADDITIONAL LITERATURE: (Not included in handbook set)		
AUTOMOTIVE HANDBOOK	231792	1-55512-125-x
INTERNATIONAL LITERATURE GUIDE (Available in Europe only)	E00029	N/A
CUSTOMER LITERATURE GUIDE	210620	N/A
MILITARY HANDBOOK (2 volume set)	210461	1-55512-126-8
SYSTEMS QUALITY/RELIABILITY	231762	1-55512-046-6

Intel Corporation is a leading supplier of microcomputer components, modules and systems. When Intel invented the microprocessor in 1971, it created the era of the microcomputer. Today, Intel architectures are considered world standards. Whether used in embedded applications such as automobiles, printers and microwave ovens, or as the CPU in personal computers, client servers or supercomputers, Intel delivers leading-edge technology.

PROGRAMMABLE LOGIC

1991

About Our Cover:

Thinkers, inventors, and artists throughout history have breathed life into their ideas by converting them into rough working sketches, models, and products. This series of covers shows a few of these creations, along with the applications and products created by Intel customers.

Intel Corporation makes no warranty for the use of its products and assumes no responsibility for any errors which may appear in this document nor does it make a commitment to update the information contained herein.

Intel retains the right to make changes to these specifications at any time, without notice.

Contact your local sales office to obtain the latest specifications before placing your order.

The following are trademarks of Intel Corporation and may only be used to identify Intel Products:

287, 376, 386, 387, 486, 4-SITE, Above, ACE51, ACE96, ACE186, ACE196, ACE960, ActionMedia, BITBUS, COMMputer, CREDIT, Data Pipeline, DVI, ETOX, FaxBACK, Genius, i, î, i486, i750, i860, ICE, iCEL, ICEVIEW, iCS, iDBP, iDIS, I²ICE, iLBX, iMDDX, iMMX, Inboard, Insite, Intel, intel, Intel386, intelBOS, Intel Certified, Intelevision, intelligent Identifier, intelligent Programming, Intellec, Intellink, iOSP, iPAT, iPDS, iPSC, iRMK, iRMX, iSBC, iSBX, iSDM, iSXM, Library Manager, MAPNET, MCS, Megachassis, MICROMAINFRAME, MULTICHANNEL, MULTIMODULE, MultiSERVER, ONCE, OpenNET, OTP, Pro750, PROMPT, Promware, QUEST, QueX, Quick-Erase, Quick-Pulse Programming, READY LAN, RMX/80, RUPI, Seamless, SLD, SugarCube, ToolTALK, UPI, VAPI, Visual Edge, VLSiCEL, and ZapCode, and the combination of ICE, iCS, iRMX, iSBC, iSBX, iSXM, MCS, or UPI and a numerical suffix.

MDS is an ordering code only and is not used as a product name or trademark. MDS® is a registered trademark of Mohawk Data Sciences Corporation.

CHMOS and HMOS are patented processes of Intel Corp.

PAL is a registered trademark of Advanced Micro Devices.

GAL is a registered trademark of Lattice Semiconductor, Inc.

Intel Corporation and Intel's FASTPATH are not affiliated with Kinetics, a division of Excelan, Inc. or its FASTPATH trademark or products.

Additional copies of this manual or other Intel literature may be obtained from:

Intel Corporation
Literature Sales
P.O. Box 7641
Mt. Prospect, IL 60056-7641

PLD CUSTOMER SUPPORT

NEW TECHNICAL LITERATURE

By the time you receive this handbook, new technical information will be available on Intel PLDs. It may include advanced information on new devices or important new applications information for devices described in the handbook. If you want the newest information on Intel PLDs, please call the toll free Literature number listed at the front of this handbook and ask for Literature Item IG69.

EPLD HOTLINE

The Intel EPLD Technical Hotline is manned by applications personnel during normal business hours. You can leave a message during off hours or when applications personnel are already handling calls. The number (U.S. and Canada) is 1-800-323-EPLD (1-800-323-3753).

BBS

Intel has a Bulletin Board System for registered iPLS II customers to electronically transfer information. A registered user with a modem can log onto the system. The current number is (916) 985-2308. If your communication software supports file transfers, you can receive utilities, software updates, and the latest information on EPLDs via the Bulletin Board.

CUSTOMER SUPPORT

INTEL'S COMPLETE SUPPORT SOLUTION WORLDWIDE

Customer Support is Intel's complete support service that provides Intel customers with hardware support, software support, customer training, consulting services and network management services. For detailed information contact your local sales offices.

After a customer purchases any system hardware or software product, service and support become major factors in determining whether that product will continue to meet a customer's expectations. Such support requires an international support organization and a breadth of programs to meet a variety of customer needs. As you might expect, Intel's customer support is extensive. It can start with assistance during your development effort to network management. 100 Intel sales and service offices are located worldwide – in the U.S., Canada, Europe and the Far East. So wherever you're using Intel technology, our professional staff is within close reach.

HARDWARE SUPPORT SERVICES

Intel's hardware maintenance service, starting with complete on-site installation will boost your productivity from the start and keep you running at maximum efficiency. Support for system or board level products can be tailored to match your needs, from complete on-site repair and maintenance support to economical carry-in or mail-in factory service.

Intel can provide support service for not only Intel systems and emulators, but also support for equipment in your development lab or provide service on your product to your end-user/customer.

SOFTWARE SUPPORT SERVICES

Software products are supported by our Technical Information Service (TIPS) that has a special toll free number to provide you with direct, ready information on known, documented problems and deficiencies, as well as work-arounds, patches and other solutions.

Intel's software support consists of two levels of contracts. Standard support includes TIPS (Technical Information Phone Service), updates and subscription service (product-specific troubleshooting guides and; *COMMENTS Magazine*). Basic support consists of updates and the subscription service. Contracts are sold in environments which represent product groupings (e.g., iRMX® environment).

NETWORK SERVICE AND SUPPORT

Today's broad spectrum of powerful networking capabilities are only as good as the customer support provided by the vendor. Intel offers network services and support structured to meet a wide variety of end-user computing needs. From a ground up design of your network's physical and logical design to implementation, installation and network wide maintenance. From software products to turn-key system solutions; Intel offers the customer a complete networked solution. With over 10 years of network experience in both the commercial and Government arena; network products, services and support from Intel provide you the most optimized network offering in the industry.

CONSULTING SERVICES

Intel provides field system engineering consulting services for any phase of your development or application effort. You can use our system engineers in a variety of ways ranging from assistance in using a new product, developing an application, personalizing training and customizing an Intel product to providing technical and management consulting. Systems Engineers are well versed in technical areas such as microcommunications, real-time applications, embedded microcontrollers, and network services. You know your application needs; we know our products. Working together we can help you get a successful product to market in the least possible time.

CUSTOMER TRAINING

Intel offers a wide range of instructional programs covering various aspects of system design and implementation. In just three to ten days a limited number of individuals learn more in a single workshop than in weeks of self-study. For optimum convenience, workshops are scheduled regularly at Training Centers worldwide or we can take our workshops to you for on-site instruction. Covering a wide variety of topics, Intel's major course categories include: architecture and assembly language, programming and operating systems, BITBUS™ and LAN applications.

DATA SHEET DESIGNATIONS

Intel uses various data sheet markings to designate each phase of the document as it relates to the product. The marking appears in the upper, right-hand corner of the data sheet. The following is the definition of these markings:

Data Sheet Marking	Description
Product Preview	Contains information on products in the design phase of development. Do not finalize a design with this information. Revised information will be published when the product becomes available.
Advanced Information	Contains information on products being sampled or in the initial production phase of development.*
Preliminary	Contains preliminary information on new products in production.*
No Marking	Contains information on products in full production.*

*Specifications within these data sheets are subject to change without notice. Verify with your local Intel sales office that you have the latest data sheet before finalizing a design.

Intel PLDs

Order Number: 296832-001

Table of Contents

Table of Contents (Continued)

Alphanumeric Index

Overview 1

October 1990

Programmable Logic Overview

Order Number: 296032-006

PROGRAMMABLE LOGIC OVERVIEW

CONTENTS

INTRODUCTION TO INTEL'S PLD FAMILY

In 1989, Intel introduced a new line of high-speed Microcomputer Programmable Logic Devices (μPLDs) aimed at augmenting high-performance microcomputer environments. This new family of devices provides high-speed support logic for the fast microcomputer systems of today. At the same time, our advanced CMOS process allows our PLDs to significantly reduce power consumption and system heat dissipation, problems that plagued the previous generation of bipolar PLDs.

Major benefits of Intel's μPLD family are:

Industry's Highest Speed PLDs	• High Speed — 80 MHz+ Registered — <10 ns t_{PD}
Specially Designed Register and Output Circuits	• Lowest Noise CMOS for Simplified High-Speed System Design • Best Solution for Metastable Conditions
Superset of Popular CMOS and Bipolar PLDs	• Compatible with Existing Design Methods • Fewer Manufacturing Line Items
Built on Standard CHMOS EPROM Technology	• High Reliability/Quality — Lower Failure Rates — 100% Programming Yields

Intel's high-performance μPLDs, along with the existing EPLD family of devices, can be classified as "User-Defined Logic" circuits. User-defined logic circuits allow system designers to tailor building block solutions to their individual systems requirements. This customization provides the needed performance, reliability, and space reduction as well as design security.

This document discusses the reasons for the trend to user-defined logic devices, briefly describes some implementation alternatives, and provides detail on a solution that can be implemented completely by the user, i.e., the programmable logic device. Details on Intels PLD product line, including terminology, nomenclature, architectural features, and developmental tools, are also described in this document.

WHY USER DEFINED LOGIC?

System designers prefer user customized ICs for the following reasons:

a. SMALLER SYSTEM SIZES: Customized components allow for reducing chip count and saving board space, resulting in smaller system physical dimensions.

b. LOWER SYSTEM COSTS: When custom LSI or VLSI components are used instead of standard SSI and MSI logic elements, there is a considerable saving in component cost per system, assembly and manufacturing cost, printed circuit board area and board costs and inventory costs.

c. HIGHER PERFORMANCE: Reduced number of ICs contributes to faster system speeds as well as lower power consumption.

d. HIGHER RELIABILITY: Since probability of failure is directly related to the number of ICs in the system, a system composed of customized LSI & VLSI chips is statistically much more reliable than the identical system made up of SSI/MSI devices.

e. DESIGN SECURITY: Systems designed with standard components can be replicated relatively easily whereas systems that contain user customized ICs cannot be copied because "reverse engineering" of the customized components is extremely difficult. Thus, use of customized ICs allows for the protection of proprietary designs.

f. INCREASED FLEXIBILITY: Customized components allow for the tailoring of systems to the end user's specific needs relatively easily. This also allows for upgradability and obsolescence protection.

USER DEFINED IC— IMPLEMENTATION ALTERNATIVES

Currently, the choices available to the system designer for customization of ICs (see Figure 1) are as follows:

(1) user programmable ICs—programmable logic devices

(2) mask programmable ICs—gate arrays

(3) standard cell based ICs

(4) full custom ICs

Alternatives (1) & (2) are usually called 'Semicustom' because in these methods only a few (less than three) of the mask layers involved in the manufacture of the IC, are customized to the users' specifications. The later two alternatives (3) & (4), involve customization of all mask layers required to manufacture the ICs to the users' specifications and are therefore called 'Custom'.

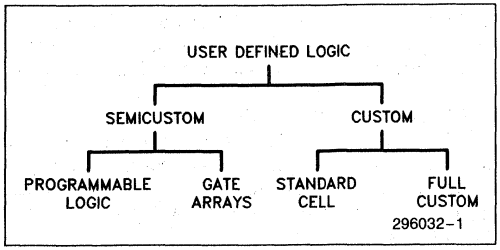


Figure 1. User-Defined Logic Implementation Choices

PROGRAMMABLE LOGIC

Most user Programmable Logic Devices (PLD) are internally structured as variations of the PLA (programmable logic array) architecture, that is composed of an array of 'AND' gates connected to an array of 'OR' gates (see Figure 2). Programmable logic devices make use of the fact that any logic equation can be converted to an equivalent 'Sum-of-Products' form and can thus be implemented in the 'AND' and 'OR' architecture. This basic PLA structure has been augmented in most PLDs with input and output blocks containing registers, latches and feedback options, that let the user implement sequential logic functions in addition to combinational logic.

The number and locations of the programmable connections between the 'AND' and 'OR' matrices as well as the input and output blocks are predetermined by the architecture of the PLD. The user, depending on his logic requirements, determines which of these connections he would like to remain open and which he would like to close, through the programming of the PLD. Programmability of these connections is achieved using various memory technologies such as fuses, EPROM cells, EEPROM cells or Static RAM cells (see Figure 3).

User programmability allows for instant customization, very similar to user programmable memories such as PROMs or EPROMs. The user can purchase a PLD off-the-shelf, use a development system running on a personal computer and, in a matter of a few hours, have customized silicon in his hands. Figure 4 compares user-defined logic alternatives.

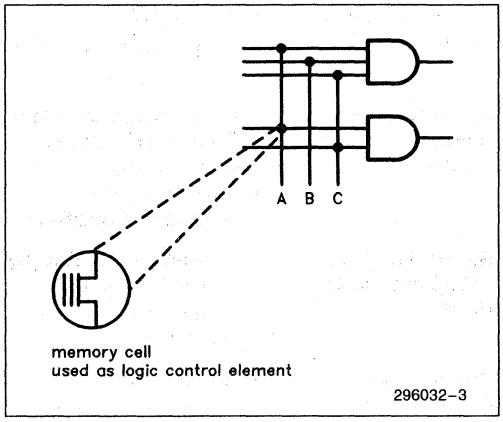


Figure 3. Programmable Connections

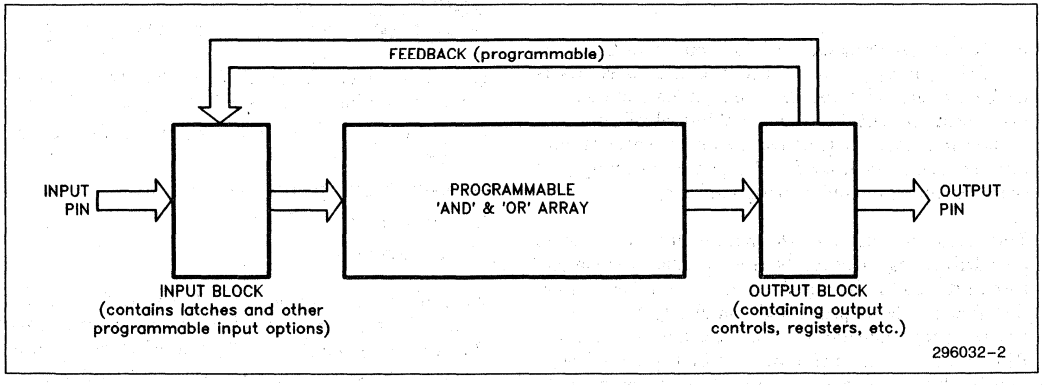


Figure 2. General Architecture of a PLD

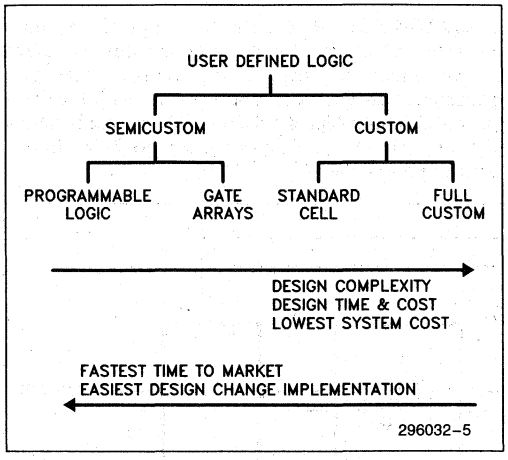


Figure 4. User-Defined Logic Alternatives Compared

LIMITATIONS OF BIPOLAR FUSE TECHNOLOGY FOR PROGRAMMABLE LOGIC DEVICES

Until 1985, all PLDs were built using Bipolar fuse technology. The bipolar fuse based devices, although offering the users the benefits of quick time to market and low development costs, had several inherent limitations.

a. HIGH POWER CONSUMPTION: Bipolar processes by nature are power hungry and as a consequence also make for very hot systems, often requiring cooling aids such as heat sinks and fans. They also cannot operate at lower voltages (2–3V) and have a lower level of noise immunity than MOS devices.

b. LOWER INTEGRATION: A fuse takes up a large amount of silicon area; this fact in conjunction with the large power requirements makes for smaller levels of integration.

c. ONE-TIME PROGRAMMABILITY: Bipolar fuses can only be blown once and cannot be reprogrammed. This does not allow for easy prototyping and could result in significant losses when preprogrammed parts are inventoried and design changes occur.

d. TESTABILITY: Since fuses can only be blown once, bipolar PLDs can only be destructively tested. Thus, testing is usually done by sampling or through additional testing elements incorporated in the chips, which can be blown to examine electrical characteristics. However, such testing methods never allow for 100% testability of all parts shipped. Thus, most users of bipolar programmable logic devices resort to extensive post-programming testing, specific to their applications.

ERASABLE PROGRAMMABLE LOGIC DEVICES

Erasable programmable logic devices (EPLD) result from the matching of CHMOS EPROM technology with the architectures of programmable logic devices. EPLDs use EPROM cells as logic control elements and therefore, when housed in windowed ceramic packages, can be erased with UV light and reprogrammed. Figure 5 shows the architecture of Intel EPLDs.

Other than the obvious benefit of reprogrammability, EPLDs offer several very significant benefits over bipolar PLDs. These are:

1. LOW POWER CONSUMPTION: Due to the CMOS technology, these products consume an order of magnitude less power than the equivalent bipolar devices. This allows for the design of complete CMOS systems, that can operate at lower voltages (less than 5V). Also, this makes for cooler systems that do not require cooling systems like fans.

2. GREATER LOGIC DENSITY: EPROM cells are an order of magnitude smaller than the smallest fuses. This means that the same function can be accommodated in significantly smaller die area, or that greater amounts of logic can now be incorporated on a single chip. Thus higher integration programmable logic devices result with the use of EPROM elements.

3. TESTABILITY: Since the EPROM cells are erasable, the entire EPROM array of the EPLD can be 100% factory tested. Thus, before the part is shipped to the customers, it can be completely tested by the programming and erasure of all the EPROM logic control bits. This testing is therefore independent of any application, in contrast to the bipolar PLDs that need application specific testing.

4. ARCHITECTURAL ENHANCEMENTS: The inherent testability of the EPROM elements allows for

significant architectural improvements over bipolar PLDs. New features, such as buried registers, programmable registers, programmable clock control, etc., can now be incorporated because of this testability. These new features allow for greatly increased utilization of the EPLDs and use of these devices in newer applications.

5. DESIGN SECURITY: EPLDs are provided with a 'security bit,' which when programmed does not allow anyone to read the programmed pattern. The logic programmed in an EPLD cannot be seen even if the die is examined (unlike bipolar PLDs—a blown fuse is clearly visible) as the stored charges are captured on a buried layer of polysilicon.

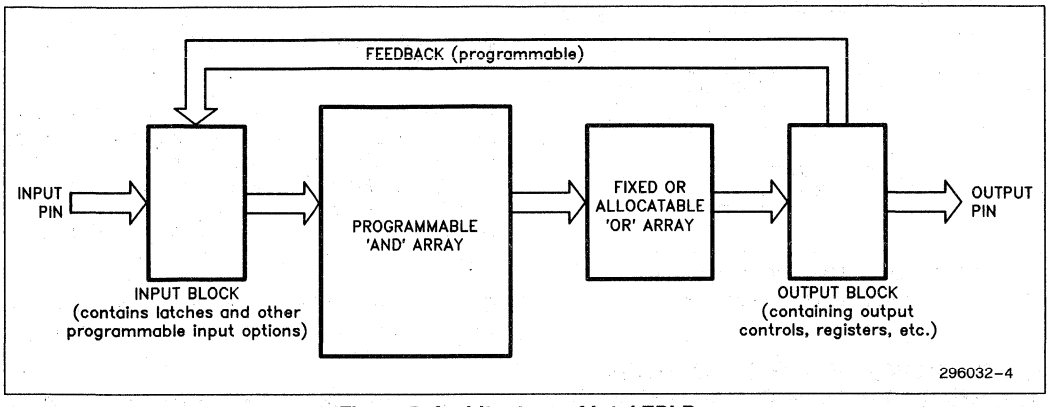


Figure 5. Architecture of Intel EPLDs

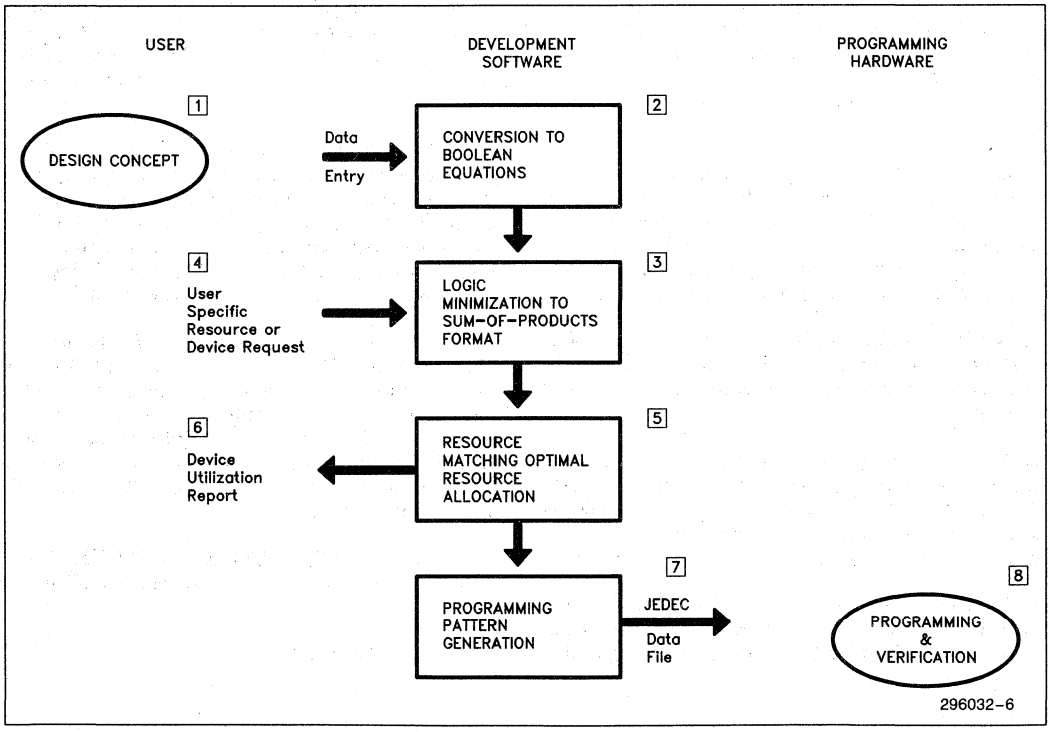


Figure 6. The PLD Design Process

The steps in a generalized design process of programmable logic is shown in Figure 6 and described in the following paragraphs.

STEP 1: The user decides on the logic he wants implemented in the PLD and enters the design into the PC or workstation. This **Design Entry** may be done by the following methods: (i)**SCHEMATIC CAPTURE**—A 'Mouse' or some other graphics input device is used to input schematics of the logic, (ii)**NET LIST ENTRY**—If the user has a hand drawn schematic he can enter the design into the computer by describing the symbols and interconnections in words using a standardized format called a net list (without using a graphics input device), (iii)**STATE EQUATION/DIAGRAM ENTRY**—Entry of a sequential design involving states and transitions between states. In the state diagram method circles represent states and the arrows interconnecting them represent the transitions. Equations or a state table can also be used to define a state machine, and (iv)**BOOLEAN EQUATIONS**—this is the most common design entry method. The logic is described in boolean algebraic equations.

STEP 2: The software converts all design entry data into boolean equations.

STEP 3: The boolean equations entered are converted to the sum of products format after logic reduction (minimization of the logic through heuristic algorithms).

STEP 4: The user has the ability to choose the PLD he would like the design implemented on. He can enter device choice and/or he can also enter in specific choices on the device as regards pinout he would like etc . . .

STEP 5: The software optimizes the logic equations to fit into the device using the minimum amount of resources (resources are input pins, output pins, registers and product terms and macrocells). This step is where the user requirements as regards required pins are taken into account. The user requests are viewed as constraints during the optimization process.

STEP 6: The software, at the end of the resource optimization/allocation, produces a report detailing the resources used up in fitting the design on the PLD. This report allows the user to incrementally stuff in logic by going back to Step 1 from this stage. Also, if the design overflowed the PLD, i.e., did not fit in the user chosen device, the software lists out the resources needed to complete the fit. The requirements such as four more inputs, one register more and one more output (are needed to complete the design) gives the user data in choosing a bigger PLD or in partitioning the intial design to fit into two devices.

STEP 7: The next step is to generate the appropriate programming pattern for the PLD. This is a standard "JEDEC" format interface and allows the output of the design software to be compatible with any piece of PROM programming hardware.

STEP 8: PROM programmer is used to program the pattern stored in the JEDEC file onto the PLD. Also, at this stage fuse programmed PLDs (bipolar) are functionally tested using test vectors included in the JEDEC file information.

CHMOS TECHNOLOGY IN EPLDs

EPLDs are manufactured with Intel's proprietary CHMOS (Complementary High-Performance MOS) technology. The backbone of the process is the integration of both a P and an N channel MOS transistor on the same substrate. In addition, EPLD's programmable architecture makes use of Intel's proven EPROM cell for programmable array interconnections as well as macrocell configuration bits. These cells are programmed electrically and erased with ultraviolet light. For details on Intel's CHMOS technology and EPROM cells technology, refer to the *Components Quality/Reliability Handbook*, Order Number 210997.

CHMOS DESIGN GUIDELINES

Designing with Intel EPLDs is relatively straightforward if the following guidelines are observed:

- Minimize the occurrence of ESD (electro-static discharge) when storing or handling EPLDs.
- Observe good design rules in printed circuit board layout.
- Provide adequate decoupling capacitance at both the device and the board level.
- Connect all unused inputs to V_{CC} or GND (CHMOS inputs should not be left floating).

Electrostatic Discharge

The two most common sources of electrostatic discharge are the human body and a charged environment.

A charged human body that touches a device lead discharges electricity into the device. Electrostatic discharge from people handling devices has long been recognized by manufacturers and users of all MOS products. Human body static electricity can be controlled by using ground straps and anti-static spray on carpeted floors. CHMOS devices should also be stored and carried in conductive tubes or anti-static foam to minimize exposure to ESD from people.

Discharge also occurs when an integrated circuit is charged to one potential and then contacts a conductor at another potential. This type of ESD can be reduced

by grounding all work surfaces, grounding all handling equipment, removing static generators such as paper from the work area, and erasing EPLDs in metal tubes, metal trays, or conductive foam.

PCB Layout

The best PCB performance is obtained when close attention is payed to V_{CC}, GND, and signal traces. V_{CC} and GND should be gridded to minimize inductive reactance and to approximate a trace layer. Clocks should be layed out to minimize crosstalk. Ensure adequate power supply and ground pins on the board connector.

Decoupling

Decouple each EPLD with a high-frequency ceramic capacitor in the range of 0.01 to 0.2 μF, depending on board frequency and current consumption. For most applications, a 0.1 μF capacitor will suffice. The following equation produces the exact value:

$$C = \frac{\Delta I_{CC}}{\Delta V / \Delta T}$$

where C = capacitor value

ΔI_{CC} = maximum switched current

ΔV = switching level

ΔT = switching time

For boards that contain mixed logic (EPLDs and TTL), observe both EPLD and TTL decoupling practices.

Unused Inputs

To minimize noise receptivity and power consumption, all unused inputs to EPLDs should be connected to V_{CC} or GND. By default, iPLS II software assigns unused inputs to GND. These pins, shown on the pinout representation of the iPLS II report file, should be connected to ground on the PCB. Pins listed as RESERVED on the report file must be left floating. Pins marked N.C. have no internal device connections and can also be left floating.

BOOLEAN MINIMIZATION TECHNIQUES FOR SOP ARCHITECTURES

Minimization plays an important role in logic design. Methods for minimization can be grouped into two classes. Class 1 includes manual methods for minimization, such as Boolean reduction or Karnaugh mapping. Class 2 is computer-assisted minimization.

Tabular methods like Karnaugh maps are efficient up to a certain point. Past that point, however, computer-assisted minimization plays a crucial part in efficient design. Even at the computer-assisted stage, the choice of minimizer software has an impact on time and the confidence level of the reduced equation (i.e., is it in the smallest possible form).

iPLS II software includes a minimizer that uses the ESPRESSO algorithms. ESPRESSO was developed by U.C. Berkeley during the summers of 1981 and 1982 in an effort to study the various strategies used by the MINI logic minimizer developed by IBM , [HON 74] and PRESTO developed by D. Brown [BRO 81]. ESPRESSO uses many of the core principles in MINI and PRESTO while improving on the speed and efficiency of their algorithms.

The primary advantage of the ESPRESSO minimizer becomes apparent when designing large finite state machines or complex, product-term intensive logic designs. In these cases, ESPRESSO arrives at the minimize solution sooner, and frequently reduces the logic to a smaller number of product terms. In certain cases where other CAD packages such as ABEL™ (PRESTO) or CUPL™ minimize equations to greater than 8 product terms, iPLS II further reduces these equations to allow the design to fit into devices supporting up to 8 product terms.

For more information on ESPRESSO, refer to *Logic Minimization Algorithms for VLSI Synthesis*, Brayton, Hachtel, McMullen, and Sangiovanni-Vincentelli, Kluwer Academic Publishers.

References

[BRO 81] D.W. Brown, "A State-Machine Synthesizer—SMS", Proc. 18th Design Automation Conference, pp. 301–304. Nashville, June 1981.

[HON 74] S. J. Hong, R. G. Cain and D. L. Ostapko, "MINI: A heuristic approach to logic minimization." *IBM Journal of Research and Development*, Vol. 18, pp. 443–458, September 1974.

LOGIC REFRESHER COURSE

Minimization of EPLD logic equations is normally performed by sophisticated algorithms that eliminate the need for tedious manual reductions. The sections provided here contain logic reference tables for cases where manual reduction techniques may be desirable.

Boolean Algebra

The Sum-of-Product architecture used in EPLDs makes Boolean algebra ideal for design analysis. The following tables summarize standard Boolean functions.

Properties

$A * B = B * A$		Commutative Property
$A + B = B + A$		
$A * (B * C) = (A * B) * C$		Associative Property
$A + (B + C) = (A + B) + C$		
$A * (B + C) = A * B + A * C$		Distributive Property
$A + B * C = (A + B) * (A + C)$		

Postulates

$0 * 0 = 0$	$0 + 0 = 0$	$\overline{0} = 1$
$0 * 1 = 0$	$0 + 1 = 1$	$\overline{1} = 0$
$1 * 1 = 1$	$1 + 1 = 1$	

Theorems

$A * 0 = 0$	$A + 0 = A$	$\overline{\overline{A}} = A$
$A * 1 = A$	$A + 1 = 1$	
$A * A = A$	$A + A = A$	
$A * \overline{A} = 0$	$A + \overline{A} = 1$	

DeMorgan's Theorems

$(\overline{A + B + C + D})$	=	$\overline{A} * \overline{B} * \overline{C} * \overline{D}$
$(\overline{A * B * C * D})$	=	$\overline{A} + \overline{B} + \overline{C} + \overline{D}$

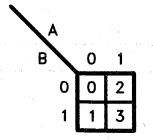
Logic Functions

$A * A$	=	A **AND** A
$A + A$	=	A **OR** A
$\overline{A}$	=	A **NOT**
$A \oplus B$ = A **EXCLUSIVE OR** B	=	$A\overline{B} + \overline{A}B$

Karnaugh Maps

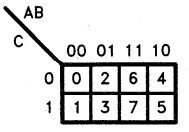
Graphical representation of data is usually easier to analyze than strings of ones and zeros. The Karnaugh Map techniques take advantage of this capability and provide an important tool to the logic designer.

Two Variables



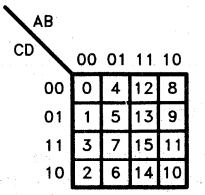
296032-7

Three Variables



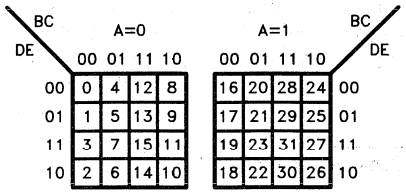
296032-8

Four Variables



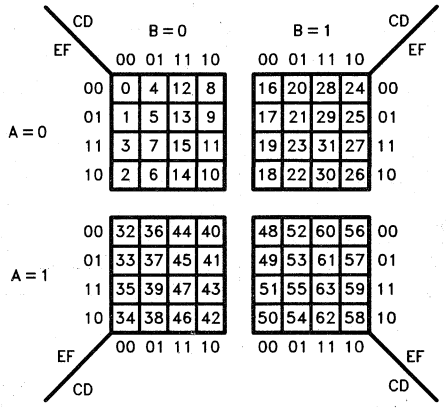
296032-9

Five Variables



296032-10

Six Variables



296032-11

Flip-Flop Tables

This subsection includes truth tables and excitation tables for the flip-flops supported by EPLDs.

D Truth Table

D	Q_N	Q_{N+1}
0	0	0
0	1	0
1	0	1
1	1	1

D Excitation Table

Q_N	Q_{N+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

T Truth Table

T	Q_N	Q_{N+1}
0	0	0
0	1	1
1	0	1
1	1	0

T Excitation Table

Q_N	Q_{N+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

JK Truth Table

J	K	Q_N	Q_{N+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

JK Excitation Table

Q_N	Q_{N+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

SR Truth Table

S	R	Q_N	Q_{N+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	Illegal	

JK Excitation Table

Q_N	Q_{N+1}	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

NOTES:
Q_N = Present State
Q_{N+1} = Next State
X = Don't Care

AUTOMATIC STANDBY MODE (TURBO BIT)

Most Intel EPLDs contain a programmable bit, the Turbo Bit, that optimizes devices for speed or power savings. When TURBO = ON, EPLDs are optimized for speed. When TURBO = OFF, they are optimized for power savings by automatically entering standby mode when input or I/O transitions are not detected over a short period of time. The following paragraphs describe how the Turbo Bit affects power and speed in EPLDs.

Turbo Off (Low Power)

Most Intel EPLDs contain circuitry that monitors inputs and feedbacks for transitions. When a transition is detected while the device is in standby mode, the circuit generates an active pulse. The leading edge of this pulse wakes the device up and the device responds according to its programming, changing outputs as necessary. If no new transitions occur during the active pulse, the device enters standby mode again. Outputs are always held valid in standby mode. Input transitions that occur during the active mode interval retrigger the active pulse. The active pulse is different depending on the device (5C060, 5AC312, etc), but is typically several times the propagation delay for a particular device.

In applications with infrequent input transitions, standby mode can result in significant power savings (see the appropriate data sheet for standby power vs. active power). The slight speed loss associated with waking up a device is in the range of 0–30 ns, which is small enough to allow standby mode to be used with most applications (see the appropriate data sheet for effect of Turbo Bit on performance).

Turbo On (Faster Speed)

In cases where the slight speed loss associated with waking a device from standby mode cannot be traded off to save power, the Turbo bit can be enabled for maximum speed operation. With the Turbo Bit enabled, the device is always in active mode, thus avoiding the wakeup delay. Note that data sheet performance is specified with the Turbo Bit enabled.

The Turbo Bit is enabled/disabled via a TURBO = ON or TURBO = OFF statement in an iPLS II ADF OPTIONS: statement. It can also be enabled/disabled by editing the JEDEC file using device programming software. With TURBO = ON the device will be programmed for high speed; with TURBO = OFF the device will be programmed for automatic standby (power savings). The default (erased) state is OFF.

PAL*/GAL* TO PLD REPLACEMENT

Already in wide use throughout the electronics industry are numerous different Programmable Logic Devices. Most common PALs and GALs can be replaced or upgraded with the following Intel PLDs:

85C220

The 85C220 is a direct, drop-in replacement for most 20-pin PALs/GALs, although some PALs have an incompatible architecture. The 85C220 runs at 80 MHz with external feedback.

85C224

The 85C224 is a direct, drop-in replacement for most 24-pin PALs/GALs, although some PALS have an incompatible architecture. The 85C224 runs at 80 MHz with external feedback.

85C060

The 85C060 is NOT a drop-in replacement for any 24-pin bipolar PAL, though it can functionally replace many higher-density devices. Some modification of CLK and OE signals may be required.

85C220 As a 20-Pin PAL Replacement

100% Compatible	
10H8, -2	16R6A
12H6, -2	16R4A
14H4, -2	16L8A
16H2, -2	16RP6A
10L8, -2	16RP4A
12L6, -2	16P8A
16L8, A-2, A-4	16R8A
16R4, A-2, A-4	16RP8A
14L4, -2	16V8A
16L2, -2	18P8
16R8, A-2, A-4	18V8
16R6, A-2, A-4	
16P8, -2	
16RP8, -2	
16RP6, -2	
16RP4, -2	
16V8	

85C224 As a 24-Pin PAL Replacement

100% Compatible	
14L8	20L8A
16L6	20R8A
18L4	20R6A
20L2	20R4A
20L8	20V8
20R8	
20R6	
20R4	

85C060 As a 24-Pin PAL Replacement

Modified Replacement
20RA10
22V10
32V10
26V10
26V12

*PAL is a registered trademark of Advanced Micro Devices.
*GAL is a registered trademark of Lattice Semiconductor, Incorporated.

QUALITY/RELIABILITY

Intel EPLDs meet the same Quality and Reliability standards as Intel's microprocessors and memories. Reliability is not just tested, but is designed into each component Intel manufactures. This assures you that Intel EPLDs will meet your system's quality/reliability needs through its life.

The methods used to guarantee the quality and reliability of Intel EPLDs parallels the methods used with Intel EPROMs. Intel's *Component Quality/Reliability Handbook*, together with *Reliability Report RR-35, EPROM Reliability Data Summary*, can provide the generic information needed to assess Intel's design and testing procedures for EPLDs. Current quality data for specific EPLDs is published in RR-64, *Intel PLD Quality/Reliability Data Summary*, printed in this handbook.

PACKAGING

Intel EPLDs are available in several packages to meet the wide requirements of customer applications. Current information on available packages is available from your local Intel field sales engineer. Detailed information on package dimensions, etc. for a particular package is provided in *Packaging Outlines and Dimensions*, Order Number 321369, which covers all Intel packages.

SOFTWARE AND PROGRAMMING SUPPORT

Intel provides design software and programming equipment to support its PLDs. In addition, Intel PLDs are supported by all major compiler and programmer manufacturers. Refer to the appendix to this handbook for detailed information.

ORDERING INFORMATION

Hotline

The Intel EPLD Technical Hotline is manned by application personnel every business day. The number for the United States and Canada is 1-800-323-EPLD (1-800-323-3753). Outside of the U.S. and Canada, contact your local Intel Sales Office. The Hotline is provided to assist with technical questions concerning Intel EPLDs. A recorder is connected for receiving messages during off-hours or when all applications personnel are busy handling calls.

BBS

Intel has a Bulletin Board System for registered iPLS and iPLS II customers to electronically transfer information. Any registered iPLS II user with a modem can log onto the system. The current number is (916) 985-2308. If your communication software supports file transfers, you can receive utilities, software updates, and the latest information on EPLDs via the Bulletin Board.

Data format for the BBS is as follows:

Start Bits: 1

Stop Bits: 1

Data Bits: 8

Speed: 300 or 1200 BAUD

Transmit/receive protocols supported are:

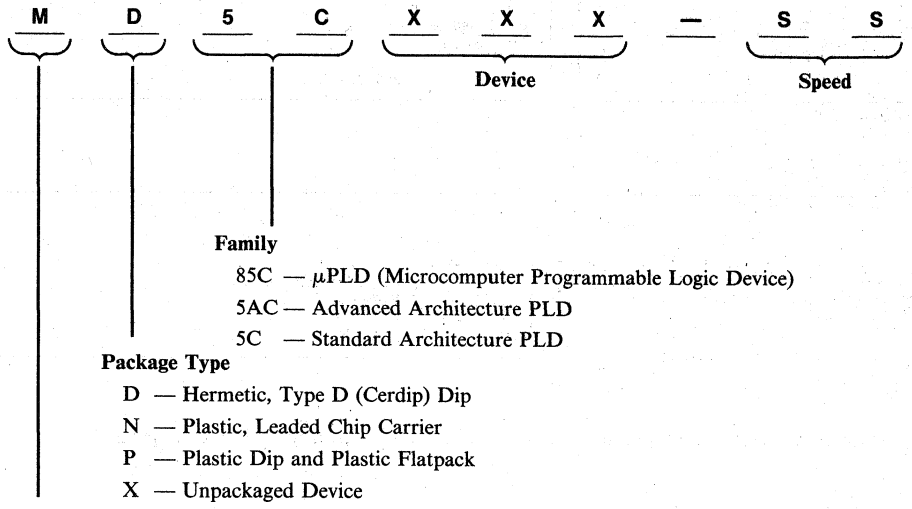
ASCII
XMODEM
KERMIT
TELINK
Cyclic Redundancy on XMODEM

EPLD Design Support

Intel has hardware designers who can help you with your EPLD designs. For more information on design assistance, contact your local Intel field sales office.

ORDERING INFORMATION (Continued)

Intel PLDs are identified as follows:



A — Indicates automotive operating temperature range (−40°C to +125°C)

J — Indicates a JAN qualified device, but is for internal identification purposes only. All JAN devices must be ordered by M38510 part number. (Example: M38510/42001 BQB), and will be marked in accordance with MIL-M-38510 specifications.

L — Indicates extended operating temperature range (−40°C to +85°C) express product with 160 + 8 hrs. dynamic burn-in.

*M — Indicates military operating temperature range (−55°C to +125°C)

Q — Indicates commercial temperature range (0°C to 70°C) express product with 160 + 8 hrs. dynamic burn-in.

T — Indicates extended temperature range (−40°C to +85°C) express product without burn-in.

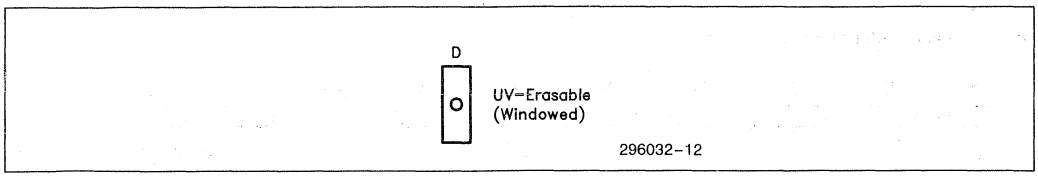
— No letter indicates commercial temperature range (0°C to 70°C) without burn-in.

Examples:

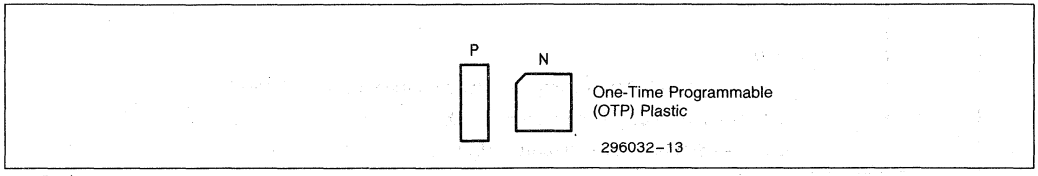
QD5C060-45 Commercial with burn-in, ceramic Dip, 060 (600 gate) device, 45 nanosecond.

*On military temperature devices, B suffix indicates MIL-STD-883C level B processing.

PLD PACKAGE TYPES



D = Windowed Ceramic DIP



P = Plastic DIP
N = Plastic PLCC

Refer to the Device Summary and each product's data sheet for available packages for a given part.

Device Data Sheets

2

μPLD	Packages	Pins	Mcells (Regs)	Dedic. Inputs	t_{PD} (ns)	f_{CNT} (MHz)	t_{SU} (ns)	t_{CO} (ns)	f_{MAX} (MHz)	I_{CC} (mA) @ Freq. (MHz)	Device Type/ Application
85C220	D, P N	20 20	8	10	10, 12	80	7	5.5	111	60 @ 80	High-speed, low power, 20-pin PAL* upgrade
85C224	D, P N	24 28	8	14	10, 12	80	7	5.5	111	60 @ 80	High-speed, low-power, 24-pin PAL* upgrade
85C060	D, P N	24 28	16	4	12, 15, 25	66	8	7	100	80 @ 66	High-speed, general purpose; 5C060/EP610/EP630 speed upgrade
85C090	D, P N	40 44	24	12	15, 20, 25	50	11	9	33.3	102 @ 50	High-speed, general purpose; 5C090/EP910/EP930 speed upgrade
85C508	D, P N	28 28	8 (latches)	16	7.5, 10	—	3 (t_{ESU})	4.5 (t_{EO})	133.3	25 @ 100	High-speed address decode/ latch; active-low outputs
85C960	D, P N	28 28	—	8	—	25	—	—	—	50 @ 50	80960 K-series bus controller; burst, wait, and ready generation control
EPLD	**Packages**	**Pins**	**Mcells (Regs)**	**Dedic. Inputs**	**t_{PD} (ns)**	**f_{CNT} (MHz)**	**t_{SU} (ns)**	**t_{CO} (ns)**	**f_{MAX} (MHz)**	**I_{CC} (mA) @ Freq. (MHz)**	**Device Type/ Application**
5AC312	D, P N	24 28	12	10	25, 30	33.3	15	15	66	100 @ 33.3	Advanced architecture, general purpose: dual-feedback, p-term allocation, Set/Reset/Oe p-terms, programmable inputs, synchronous/asynchronous clocks
5AC324	D, P N	40 44	24	12	25, 30, 35	33	12.5	17.8	66	175 @ 33	
5C032	D, P	20	8	10	30, 35, 40	25	23	17	43.4	30 @ 25	General purpose PLDs (interface logic, state machines, sequencers/ controllers)
5C060	D, P N	24 28	16	4	45, 55	16.6	38	22	26.3	95 @ 16.6	
5C090	D, P N	40 44	24	12	50, 60	16.4	38	23	26.3	150 @ 16.4	
5C180	A N	68 68	48	16	70, 75, 90	12.2	53	29	20.8	200 @ 12.2	

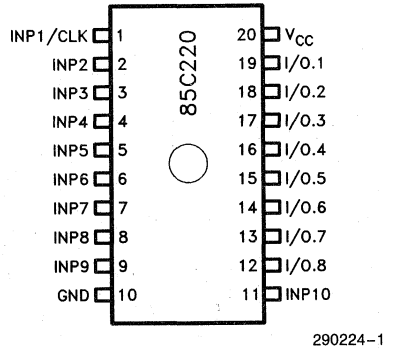
PACKAGES:
A = Pin Grid Array (Windowed for UV Erase)
P = Plastic DIP (One-Time Programmable)
D = Ceramic DIP (Windowed for UV Erase)
N = PLCC (One-Time Programmable)

*PAL is a registered trademark of Advanced Micro Devices.

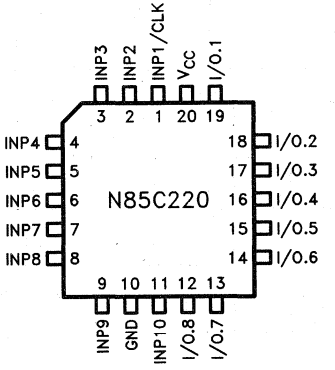
85C220
FAST 1-MICRON CHMOS 8-MACROCELL μPLD

- **High-Performance, Low-Power Upgrade for SSI/MSI Logic and Bipolar PALs* in Intel386™, i486™, i860™, 80960 Series, and Other High-Performance Systems**
- **80 MHz Max Frequency (External Feedback), Clock to Output 5.5 ns (max)**
- **Performance/Power Upgrade to "D- and E-Speed" PLDs in State Machine Applications**
- **t_{PD} 10 ns (max), 100 MHz Max Frequency (Internal Feedback), 111 MHz (Pipelined)**
- **Eight Macrocells with Programmable I/O Architecture (Register/ Combinatorial)**
- **8 P-Terms, Selectable SOP Invert, OE P-Term for Each Macrocell**
- **Up to 18 Inputs (10 Dedicated & 8 I/O) and 8 Outputs**
- **Typical I_{CC} = 35 mA at 15 MHz**
- **Emulates 20-pin PAL Architectures**
- **High-Speed Upgrade to EP320, EP330, and 5C032**
- **1-Micron CHMOS* IIIE EPROM Technology. UV-Erasable (CerDIP) or OTP**
- **Programmable Low-Power Option for "Standby" Operation; 25 μA Typical Current in Standby Mode**
- **Programmable "Security Bit" Allows Total Protection of Proprietary Designs**
- **100% Generically Tested Logic Array**
- **Available in 300-mil 20-pin CerDIP/PDIP Packages and 20-pin PLCC**
  (See Packaging Spec., Order #231369)

Intel386™, i486™ and i860™ and registered trademarks of Intel Corporation.
CHMOS is a patented process of Intel Corp.
*PAL is a registered trademark of Advanced Micro Devices.



290224-1



290224-2

Figure 1. Pinout Diagrams

October 1990
Order Number: 290224-002

INTRODUCTION

The Intel 85C220 1-micron CHMOS μPLD (Microcomputer Programmable Logic Device) is capable of implementing over 300 equivalent gates of user-customized logic functions through programming. This device can be used to upgrade high-speed bipolar programmable logic devices and 74-series LS and CMOS SSI and MSI logic devices in bus control and state-machine applications for Intel386™, i486™, and Intel i860™ based systems and other high-performance processors. The 85C220 can also be used as a direct, low-power replacement for almost all high-speed 20-pin fuse-based programmable logic devices. With its flexible I/O architecture and fast speeds, this device has functional capabilities that surpass those of typical programmable logic devices.

The 85C220 uses advanced EPROM cells as architecture and logic control memory elements. Coupled with Intel's proprietary CHMOS IIIE technology, the result is a device that offers a fast t_{PD} in combinatorial mode, with current consumption much lower than bipolar devices of equivalent speed. The maximum "count" frequency of 80 MHz is optimized for high-performance state machines typically encountered in bus control applications. EPROM technology allows these devices to be 100% factory tested by programming and erasing all the EPROM logic control elements.

The inherent speed of the device together with its lower power demands and plastic package make the 85C220 an ideal production vehicle for high-volume manufacturing of high-performance systems. For state machine applications, the 85C220 can upgrade most commonly used high-speed bipolar PLDs to improve performance while decreasing power consumption.

ARCHITECTURE DESCRIPTION

The architecture of the 85C220 is based on the SOP (Sum of Products) PAL structure with a programmable AND array feeding into a fixed OR array. Programmable macrocells allow the device to accommodate both combinatorial and sequential logic functions. Each macrocell is individually programmable for combinatorial or registered output. An invert option on the SOP allows each output to be configured as an active-high or active-low output.

As shown in Figure 2, the 85C220 contains 10 dedicated inputs and 8 I/O pins. Each I/O pin can be individually programmed to function as an input, output, or bidirectional I/O pin. Associated with each I/O pin is a programmable macrocell.

Figure 3 shows the structure of the 85C220 macrocell. Each macrocell includes a p-term (product term) block with eight AND p-terms feeding the OR gate of the I/O control block and one additional p-term controlling the output buffer. The logic array is 36 rows wide, allowing each p-term in the device to connect to the true or complement of each input and I/O feedback signal. Each intersecting point in the logic array is connected or not connected based on the value programmed in the EPROM array. Initially (EPROM erased state), all p-terms are connected to all signals. Connections are broken by programming the appropriate EPROM cells. Connecting both the true and complement of a signal for a given p-term removes that p-term from the SOP for the macrocell (i.e., that p-term is a "don't care").

Figure 4 shows the architecture of each macrocell's I/O control block. The SOP input to the I/O control block can be inverted or non-inverted. The output can be registered or combinatorial. When registered output is selected, feedback to the logic array comes directly from the register (before the output buffer). When combinatorial output is selected, feedback comes from the I/O pin (after the output buffer) and can be used for bidirectional I/O. The register is a D-type register that clocks on the rising edge of CLK.

20-PIN PLD COMPATIBILITY

The 85C220 is designed to be a logical superset of most high-speed 20-pin bipolar PAL devices. The I/O and logic sections of the device can be configured to emulate any of the devices listed below. Designers can often replace multiple PALs with fewer 85C220 devices. The following list includes some of the devices with which the 85C220 is compatible:

10L8	16L2	16L8
10H8	16H2	16P8
10P8	16P2	16R8
12L6	16R4	16R8A
12H6	16RP4	16RP8
12P6	16H6	16V8
14L4	16R6	18CV8
14H4	16RP6	
14P4	16C1	

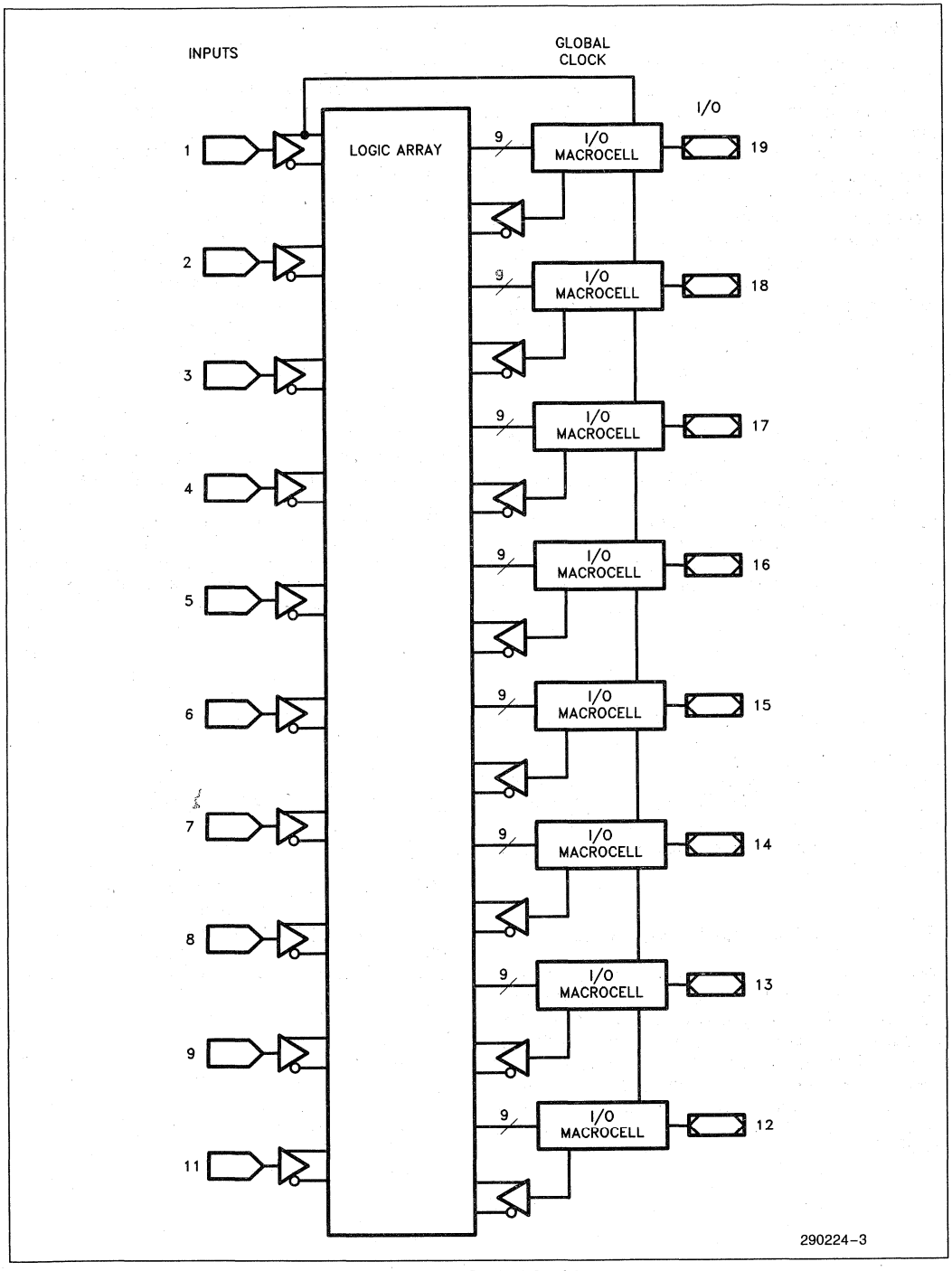


Figure 2. 85C220 Global Architecture

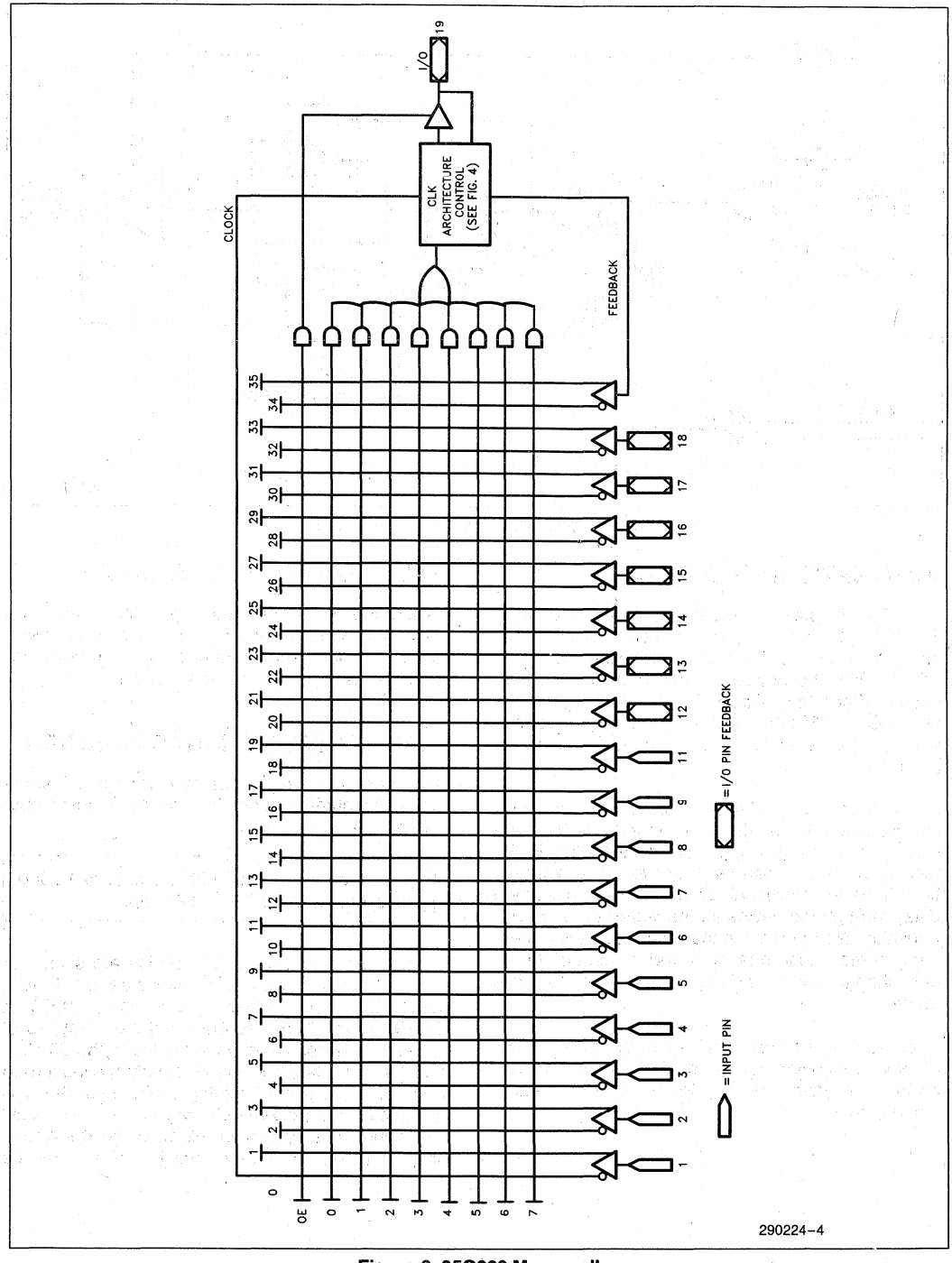


Figure 3. 85C220 Macrocell

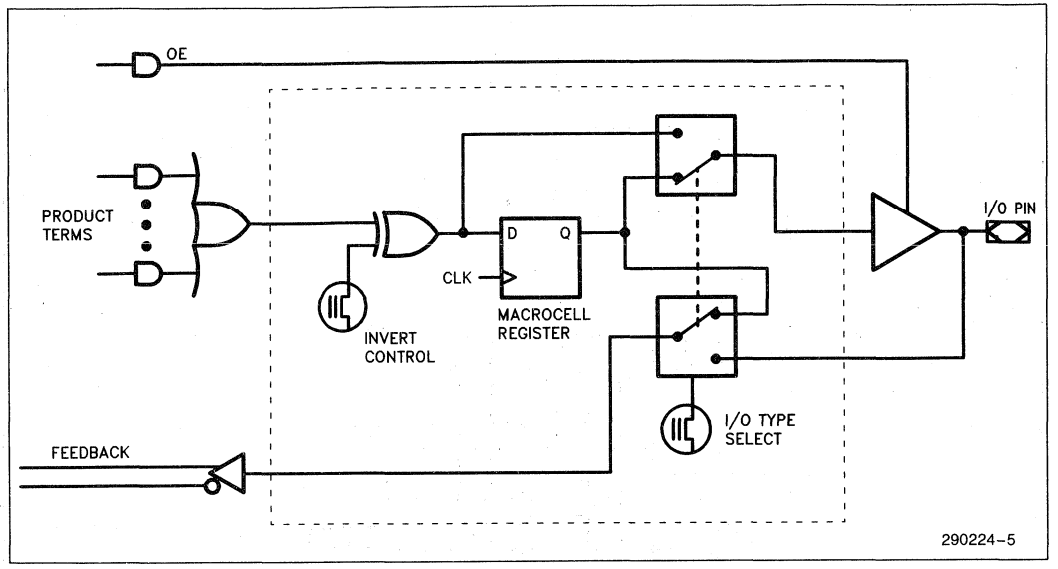


Figure 4. 85C220 I/O Control Architecture

AUTOMATIC STAND-BY MODE

The 85C220 contains a programmable bit, the Turbo Bit, that optimizes operation for speed or for power savings. When the Turbo Bit is programmed (TURBO = ON), the device is optimized for maximum speed. When the Turbo Bit is not programmed (TURBO = OFF), the device is optimized for power savings by entering standby mode during periods of inactivity.

Figure 5 shows the device entering standby mode approximately 75 ns after the last input or I/O transition. When the next input or I/O transition is detected, the device returns to active mode. Wakeup time adds an additional 20 ns to the propagation delay through the device as measured from the first transition. No delay will occur if an output is dependent on more than one input and the last of the inputs changes after the device has returned to active mode.

After erasure, the Turbo Bit is unprogrammed (OFF); automatic standby mode is enabled. When the Turbo Bit is programmed (ON), the device never enters standby mode.

POWER-ON CHARACTERISTICS

85C220 inputs and outputs begin responding 1 μs (max.) after V_{CC} power-up (V_{CC} = 4.75V) or after a power-loss/power-up sequence. All macrocells programmed as registers will be set to a logic low.

ERASED STATE CHARACTERISTICS

Prior to programming or after erasure, the I/O structure is configured for combinatorial active low output with input (pin) feedback.

Erasure time for the 85C220 is 1 hour at 12,000 μWsec/cm^2 with a 2537Å UV lamp.

Erasure characteristics of the device are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000Å. It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000Å–4000Å range. Data shows that constant exposure to room level fluorescent lighting could erase the typical 85C220 in approximately six years, while it would take approximately two weeks to erase the device when exposed to direct sunlight. If the device

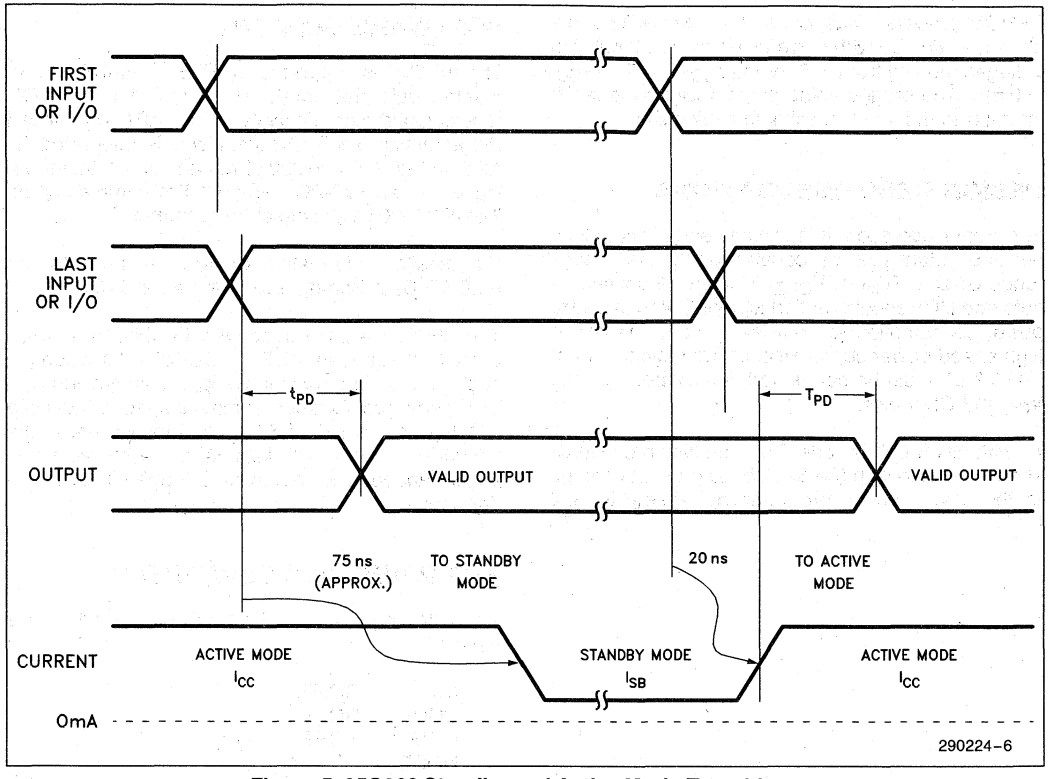


Figure 5. 85C220 Standby and Active Mode Transitions

is to be exposed to these lighting conditions for extended periods of time, conductive opaque labels should be placed over the device window to prevent unintentional erasure.

The recommended erasure procedure for the 85C220 is exposure to shortwave utraviolet light with a wavelength 2537Å. The integrated dose (i.e., UV intensity x exposure time) for erasure should be a minimum of forty (40) Wsec/cm^2. The erasure time with this dosage is approximately 1 hour using an ultraviolet lamp with a 12,000 μW/cm^2 power rating. The device should be placed within 1 inch of the lamp tubes during exposure. The maximum integrated dose the 85C220 can be exposed to without damage is 7258 Wsec/cm^2 (1 week at 12,000 μW/cm^2). Exposure to high intensity UV light for longer periods may cause permanent damage to the device.

int$_e$ligent Programming™ Algorithm

The 85C220 supports the int$_e$ligent Programming Algorithm, which rapidly programs Intel PLDs, while maintaining a high degree of reliability. It is particularly suited for production programming environments. This method ensures reliability as the incremental programming margin of each bit has been verified during programming. Programming voltage and waveform specifications are available by request from Intel to support programming the device.

LATCH-UP IMMUNITY

All of the input, output, and clock pins of the device have been designed to resist latch-up which is inherent in inferior CMOS structures. The 85C220 is designed with Intel's proprietary 1-micron CHMOS IIIE

EPROM process. Thus, each of the pins will not experience latch-up with currents up to ±100 mA and voltages ranging from −0.5V to (V_{CC} + 0.5V) (V_{CC} +0.5V). The programming pin is designed to resist latch-up to the 13.5V maximum device limit.

DESIGN RECOMMENDATIONS

For proper operation, it is recommended that all input and output pins be constrained to the voltage range GND ≤ (V_{IN} or V_{OUT}) ≤ V_{CC}. All unused inputs and I/Os should be tied high or low to minimize power consumption (do not leave them floating). A high-speed power supply decoupling capacitor of at lest 0.2 μF must be connected directly between the V_{CC} and GND pins.

As with all CMOS devices, ESD handling procedures should be used with the 85C220 to prevent damage to the device during programming, assembly, and test.

SOFTWARE SUPPORT

The 85C220 is supported by iPLS II (Intel Programmable Logic Software), which includes the LOC (Logic Optimizing Compiler) and APT (Advanced Programming Tool). Programming is supported by APT on the GUPI 20D20J Programming Adaptor using either an iUP-PC Personal Programmer or an iUP-200A/201A Universal Programmer.

For detailed information on iPLS II, refer to the IPLDS II Data Sheet, order number: 290134.

The 85C220 is also supported by third-party logic compilers such as ABEL*, CUPL*, PLDesigner*, Log/IC, etc. Programming support is provided by third-party programmer companies such as Data I/O, Logical Devices, STAG, etc. Please refer to the "Third-Party Support" lists in the *Programmable Logic* handbook for complete information and vendor contacts.

ADF PRIMITIVES SUPPORTED

The following ADF primitives are supported by this device:

INP	RONF
CONF	RORF
COIF	NORF
NOCF	

ORDERING INFORMATION

f_{CNT1} (MHz)	f_{MAX} (MHz)	t_{PD} (ns)	Order Code	Package	Operating Range
80	111	10	D85C220-80	*CerDIP	Commercial
			P85C220-80	PDIP	
			N85C220-80	PLCC	
66	90.9	12	D85C220-66	*CerDIP	Commercial
			P85C220-66	PDIP	
			N85C220-66	PLCC	

*Windowed CerDIP package allows UV erase.

*ABEL is a trademark of Data I/O, Corporation. CUPL is a trademark of Logical Devices, Inc. PLDesigner is a trademark of MINC, Inc.

ABSOLUTE MAXIMUM RATINGS*

Supply Voltage (V_{CC})(1) . . . . . −2.0V to +7.0V

Programming Supply Voltage (V_{PP})(1) . . . . . −2.0V to +13.5V

D.C. Input Voltage (V_I)(1, 2) . . . −0.5V to V_{CC} +0.5V

Storage Temperature (T_{STG}) . . . . −65°C to +150°C

Ambient Temperature (T_{AMB})(3) . . −10°C to +85°C

NOTES:
1. Voltages with respect to GND.
2. Minimum D.C. input is −0.5V. During transitions, the inputs may undershoot to −2.0V or overshoot to +7.0V for periods of less than 20 ns under no load conditions.
3. Under bias. Extended Temperature versions are also available.

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

2

RECOMMENDED OPERATING CONDITIONS

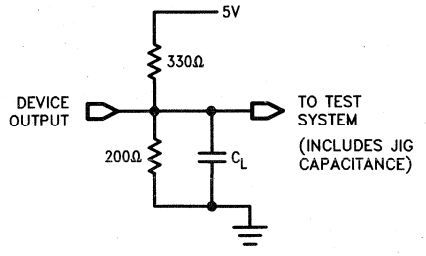
Symbol	Parameter	Min	Max	Units
V_{CC}	Supply Voltage	4.75	5.25	V
V_{IN}	Input Voltage	0	V_{CC}	V
V_O	Output Voltage	0	V_{CC}	V
T_A	Operating Temperature	0	+70	°C
t_R	Input Rise Time		500	ns
t_F	Input Fall Time		500	ns

D.C. CHARACTERISTICS (T_A = 0°C to +70°C, V_{CC} = 5.0V ±5%)

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
V_{IH}(4)	High Level Input Voltage	2.0		V_{CC}+0.3	V	
V_{IL}(4)	Low Level Input Voltage	−0.3		0.8	V	
V_{OH}	High Level Output Voltage	2.4			V	I_O = −4.0 mA D.C., V_{CC} = Min
V_{OL}(5)	Low Level Output Voltage			0.45	V	I_O = 12.0 mA D.C., V_{CC} = Min
I_I	Input Leakage Current			±10	µA	V_{CC} = Max, GND < V_{IN} < V_{CC}
I_{OZ}	Output Leakage Current			±10	µA	V_{CC} = Max, GND < V_{OUT} < V_{CC}
I_{SC}(6)	Output Short Circuit Current	−30		−120	mA	V_{CC} = Max, V_{OUT} = 0.5V
I_{SB}(7)	Standby Current		50	100	µA	V_{CC} = Max, V_{IN} = V_{CC} or GND, Standby Mode
I_{CC}	Power Supply Current (see I_{CC} vs Fred. graph)		2	5	mA	V_{CC} = Max, V_{IN} = V_{CC} or GND, No Load, f_{IN} = 1 MHz, Device Prog. as an 8-Bit Counter, Non-Turbo Mode
			35	50	mA	f_{IN} = 15 MHz, Active Mode
			45	60	mA	f_{IN} = 80 MHz, Active Mode

NOTES:
4. Absolute values with respect to device GND; all over and undershoots due to system or tester noise are included.
5. Maximum DC I_{OL} for the device (all 8 outputs) is 64 mA.
6. Not more than 1 output should be tested at a time. Duration of that test should not exceed 1 second.
7. In Non-Turbo Mode (TURBO=OFF), device enters standby mode approximately 75 ns after the last input transition.

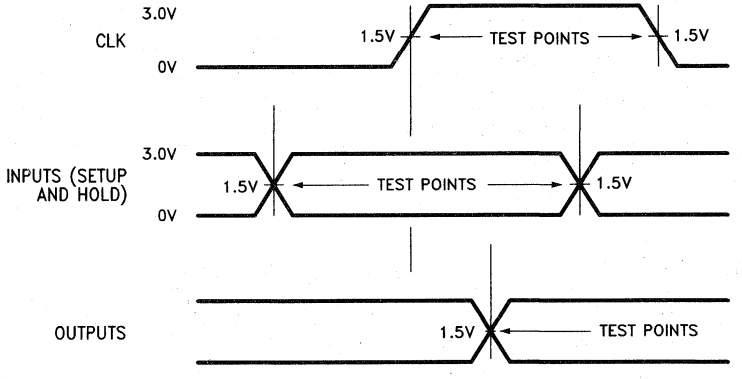
A.C. TESTING LOAD CIRCUIT



290224-7

C_L = 30 pF

A.C. TESTING WAVEFORM—SYNCHRONOUS INPUTS AND OUTPUTS



290224-8

A.C. Testing: Inputs (including CLK) are driven at 3.0V for a Logic "1" and 0V for a Logic "0". Setup and hold input measurements relative to CLK are made at 2.0V on CLK (low-to-high transition). Inputs are measured at 2.0V for a high and 0.8V for a low. Outputs are measured at 1.5V. Device input rise and fall times are less than 3 ns.

CAPACITANCE (T_A = 0°C to +70°C; V_{CC} = 5.0V ±5%)(8)

Symbol	Parameter	Min	Typ	Max	Unit	Conditions
C_{IN}	Input Capacitance		4	6	pF	V_{IN} = 0V, f = 1.0 MHz
C_{IO}	I/O Capacitance		5	8	pF	V_{OUT} = 0V, f = 1.0 MHz
C_{CLK}	CLK Capacitance		6	8	pF	V_{OUT} = 0V, f = 1.0 MHz
C_{VPP}	V_{PP} Pin Capacitance		8	10	pF	V_{PP} on Pin 11, f = 1.0 MHz

NOTES:
8. These values are evaluated during initial characterization and whenever design modifications occur that may affect capacitance.

A.C. CHARACTERISTICS (T_A = 0°C to +70°C, V_{CC} = 5.0V ±5%)[9]

Symbol	Parameter	85C220-80			85C220-66			Non-Turbo[10] Mode	Units
		Min	Typ	Max	Min	Typ	Max		
t_{PD}[11]	Input or I/O to Output	4		10	4		12	+20	ns
t_{PZX}[12]	Input or I/O to Output Enable	4		12	4		12	+20	ns
t_{PXZ}[12]	Input or I/O to Output Disable	4		10	4		12	+20	ns

NOTES:
9. Typical values are at T_A = +25°C, V_{CC} = 5V, Active Mode.
10. If device is operated in Non-Turbo Mode (TURBO=OFF) and the device is inactive for approximately 75 ns, increase time by amount shown.
11. Measured with all eight outputs switching.
12. t_{PZX} and t_{PXZ} are measured at ±0.5V from steady state voltage as driven by specification output load. t_{PXZ} is measured with C_L = 5 pF. Measured with all eight outputs switching.

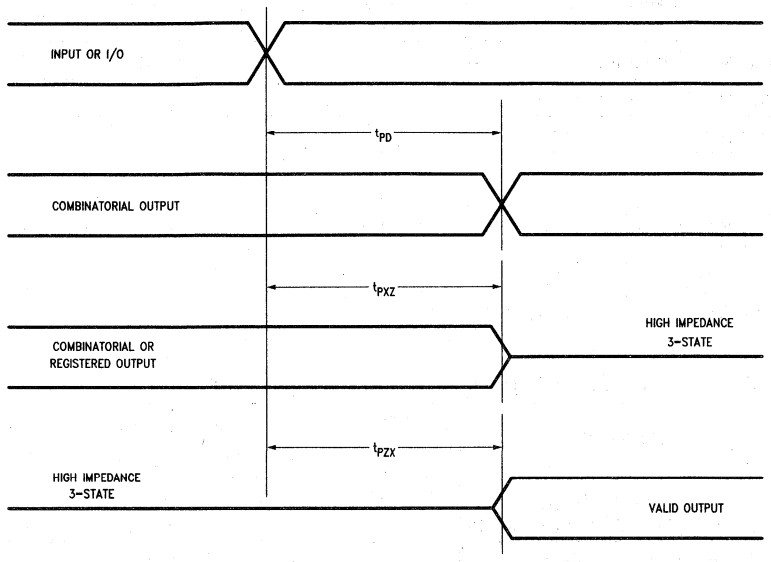
2

SYNCHRONOUS CLOCK MODE (T_A = 0°C to 70°C, V_{CC} = 5V ±5%)[9]

Symbol	Parameter	85C220-80			85C220-66			Non-Turbo[10] Mode	Units
		Min	Typ	Max	Min	Typ	Max		
f_{CNT1}[11]	Maximum Counter Frequency $1/(t_{SU} + t_{CO})$—External Feedback		100	80		80	66		MHz
f_{CNT2}[11]	Maximum Counter Frequency $1/(t_{CNT})$—Internal Feedback		111	100		90	83.3		MHz
f_{MAX}[11]	Maximum Frequency (Pipelined) $1/(t_{CW})$—No Feedback		125	111		100	90.9		MHz
t_{SU}	Input or I/O Setup Time to CLK	7			9			+20	ns
t_H	Input or I/O Hold Time from CLK	0			0				ns
t_{CO1}	CLK High to Output Valid	1.5[13]		5.5[11]	1.5[13]		6[11]		ns
t_{CO2}	CLK High to Output Valid Fed through Comb. Macrocell	4.5		13	4.5		15	+20	ns
t_{CNT}[11]	Macrocell Output Feedback to Macrocell Input—Internal Path	10			12			+20	ns
t_{CL}	CLK Low Time	4			5				ns
t_{CH}	CLK High Time	4			5				ns
t_{CW}	CLK Width	9			11				ns

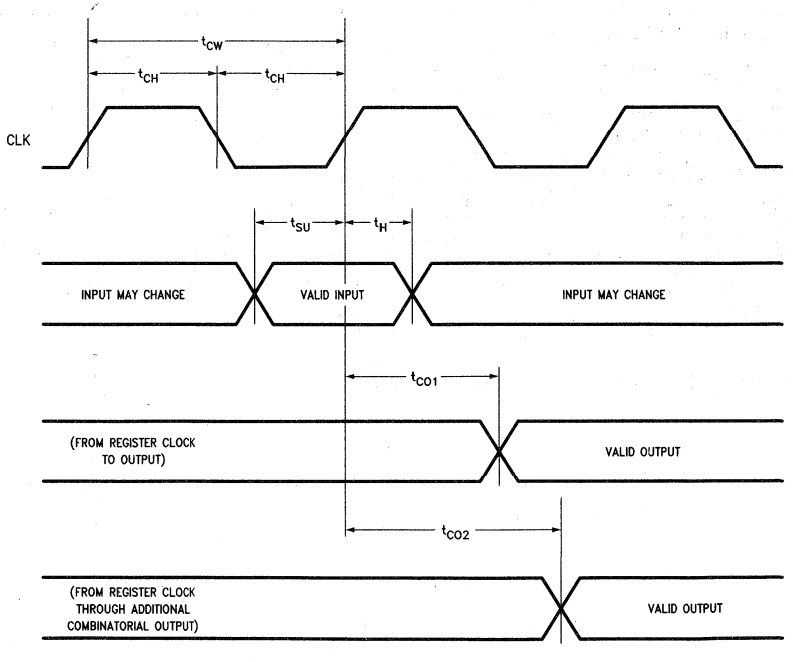
NOTE:
13. t_{CO1} min. is measured with one output switching, T_A = 0°C, V_{CC} = 5.25V.

COMBINATORIAL MODE



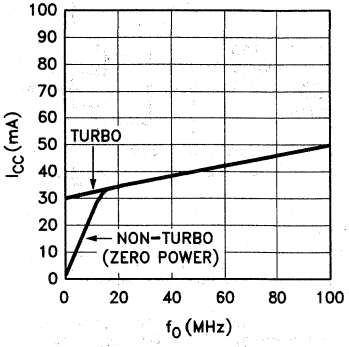
290224-9

REGISTERED MODE



290224-10

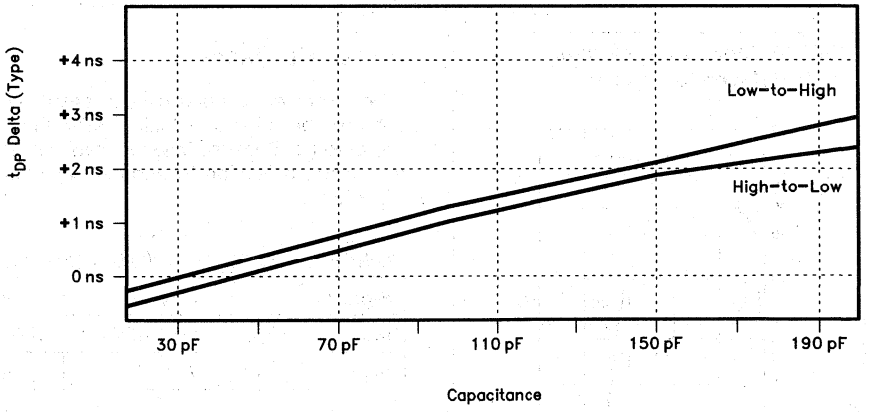
85C220
I_{CC} vs. Frequency



290224-11

Conditions:
T_A = 25°C
V_{CC} = 5.25V

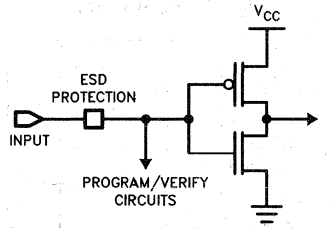
85C220
t_{PD} Derating vs. Capacitive Loading



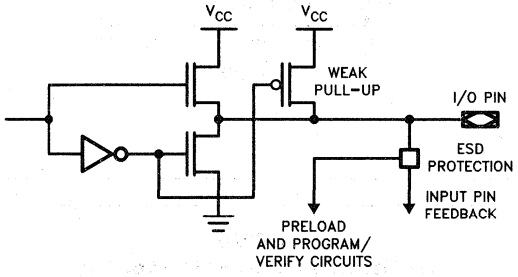
290224-12

Conditions:
T_A = 25°C
V_{CC} = 5.0V

Input/Output Equivalent Schematics

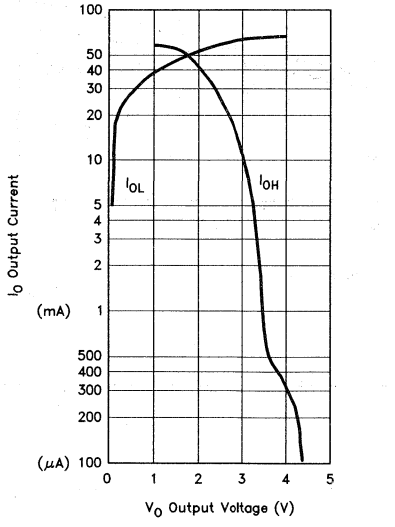


290224-13



290224-14

85C220 Output Drive Current in Relation to Voltage



290224-15

Conditions:
$T_A = +80°C$
$V_{CC} = 4.75V$

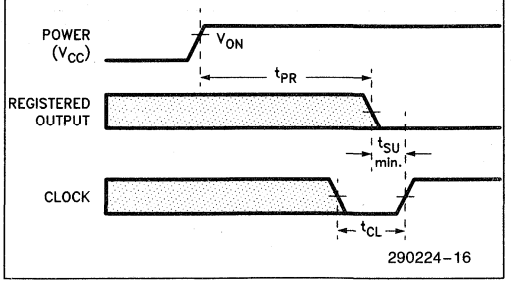
Power-Up Reset

Internal power-up reset circuits ensure that all flip-flops will be reset to a logic 0 after the device has powered up. Because V_{CC} rise can vary significantly from one application to another, V_{CC} rise must be monotonic.

POWER-UP RESET CHARACTERISTICS

Symbol	Parameter	Value
t_{PR}	Power-Up Reset Time	1000 ns Max.
V_{ON}	Turn-On Voltage	4.75V

POWER-UP RESET



290224-16

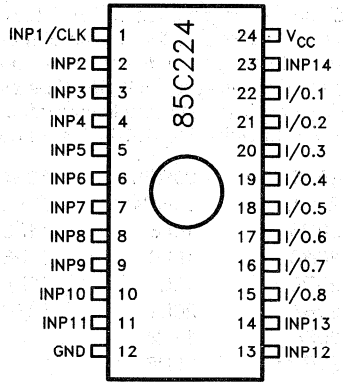
85C224
FAST 1-MICRON CHMOS 8-MACROCELL μPLD

- **High-Performance, Low-Power Upgrade for SSI/MSI Logic and Bipolar PALs* in Intel386™, i486™, i860™, 80960 Series, and Other High-Performance Systems**
- **80 MHz Max Frequency (External Feedback), Clock to Output 5.5 ns (max)**
- **Performance/Power Upgrade to "D- and E-Speed" PLDs in State Machine Applications**
- **t_{PD} 10 ns (max), 100 MHz Max Frequency (Internal Feedback), 111 MHz (Pipelined)**
- **8 Macrocells with Programmable I/O Architecture (Register/Combinatorial)**
- **8 P-terms, Selectable SOP Invert, OE P-term for Each Macrocell**
- **Up to 22 Inputs (14 Dedicated & 8 I/O) and 8 Outputs**
- **Typical I_{CC} = 35 mA at 15 MHz**
- **Emulates 24-pin PAL Architectures**
- **1-Micron CHMOS** IIIE EPROM Technology. UV-Erasable (CerDIP) or OTP**
- **Programmable Low-Power Option for "Standby" Operation; 25 μA Typical Current in Standby Mode**
- **Programmable "Security Bit" Allows Total Protection of Proprietary Designs**
- **100% Generically Tested Logic Array**
- **Available in 300-mil 24-pin CERDIP/ PDIP Packages and 28-pin PLCC**

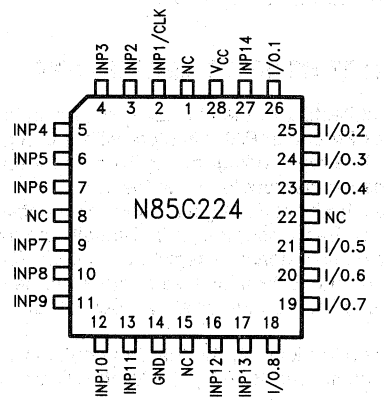
(See Packaging Spec., Order Number #231369)

Intel386, i486, and i860 are trademarks of Intel, Corp.
*PAL is a registered trademark of Advanced Micro Devices
**CHMOS is a patented process of Intel, Corp.

2



290268-1



290268-2

Figure 1. Pinout Diagrams

October 1990
Order Number: 290268-001

INTRODUCTION

The Intel 85C224 1-micron CHMOS μPLD (Microcomputer Programmable Logic Device) is capable of upgrading high-speed bipolar programmable logic devices and 74-series LS and CMOS SSI and MSI logic devices in bus control and state-machine applications for Intel386™-, i486™, and Intel i860™-based systems and other high-performance processors. The 85C224 can also be used as a direct, low-power replacement for almost all high-speed 24-pin fuse-based programmable logic devices. With its flexible I/O architecture and fast speeds, this device has functional capabilities that surpass those of typical programmable logic devices.

The 85C224 uses advanced EPROM cells as architecture and logic control memory elements. Coupled with Intel's proprietary CHMOS IIIE technology, the result is a device that offers a fast t_{PD} in combinatorial mode with current consumption much lower than bipolar devices of equivalent speed. The maximum "count" frequency of 80 MHz is optimized for high-performance state machines typically encountered in bus control applications. EPROM technology allows these devices to be 100% factory tested by programming and erasing all the EPROM logic control elements.

The inherent speed of the device together with its lower power consumption and plastic package make the 85C224 an ideal production vehicle for high-volume manufacturing of high-performance systems. For state machine applications, the 85C224 can upgrade most commonly used high-speed bipolar PLDs to improve performance while significantly decreasing power consumption.

ARCHITECTURE DESCRIPTION

The architecture of the 85C224 is based on the SOP (Sum of Products) PAL structure with a programmable AND array feeding into a fixed OR array. Programmable macrocells allow the device to accommodate both combinatorial and sequential logic functions. Each macrocell is individually programmable for combinatorial or registered output. An invert option on the SOP allows each output to be configured as an active-high or active-low output.

As shown in Figure 2, the 85C224 contains 14 dedicated inputs and 8 I/O pins. Each I/O pin can be individually programmed to function as an input, output, or bidirectional I/O pin. Associated with each I/O pin is a programmable macrocell.

Figure 3 shows the structure of the 85C224 macrocell. Each macrocell includes a p-term (product term) block with eight AND p-terms feeding the OR gate of the I/O control block and one additional p-term controlling the output buffer. The logic array is 44 rows wide, allowing each p-term in the device to connect to the true or complement of each input and I/O feedback signal. Each intersecting point in the logic array is connected or not connected based on the value programmed in the EPROM array. Initially (EPROM erased state), all p-terms are connected to all signals. Connections are broken by programming the appropriate EPROM cell. Connecting both the true and complement of a signal for a given p-term removes that p-term from the SOP for the macrocell (i.e., that p-term is a "don't care").

Figure 4 shows the architecture of each macrocell's I/O control block. The SOP input to the I/O control block can be inverted or non-inverted. The output can be registered or combinatorial. When registered output is selected, feedback to the logic array comes directly from the register (before the output buffer). When combinatorial output is selected, feedback comes from the I/O pin (after the output buffer) and can be used for bidirectional I/O. The register is a D-type register that clocks on the rising edge of the global CLK.

24-PIN PLD COMPATIBILITY

The 85C224 is designed to be a logical superset of most high-speed 24-pin bipolar PAL devices. The I/O and logic sections of the device can be configured to emulate any of the devices listed below. Designers can often replace multiple PALs with fewer 85C224 devices. The following list includes some of the devices with which the 85C224 is compatible:

20V8	20RP6	18H4
20L8	20RP4	20H2
20H8	14L8	14P8
20R8	16L6	16P6
20R6	18L4	18P4
20R4	20L2	20P2
20P8	14H8	
20RP8	16H6	

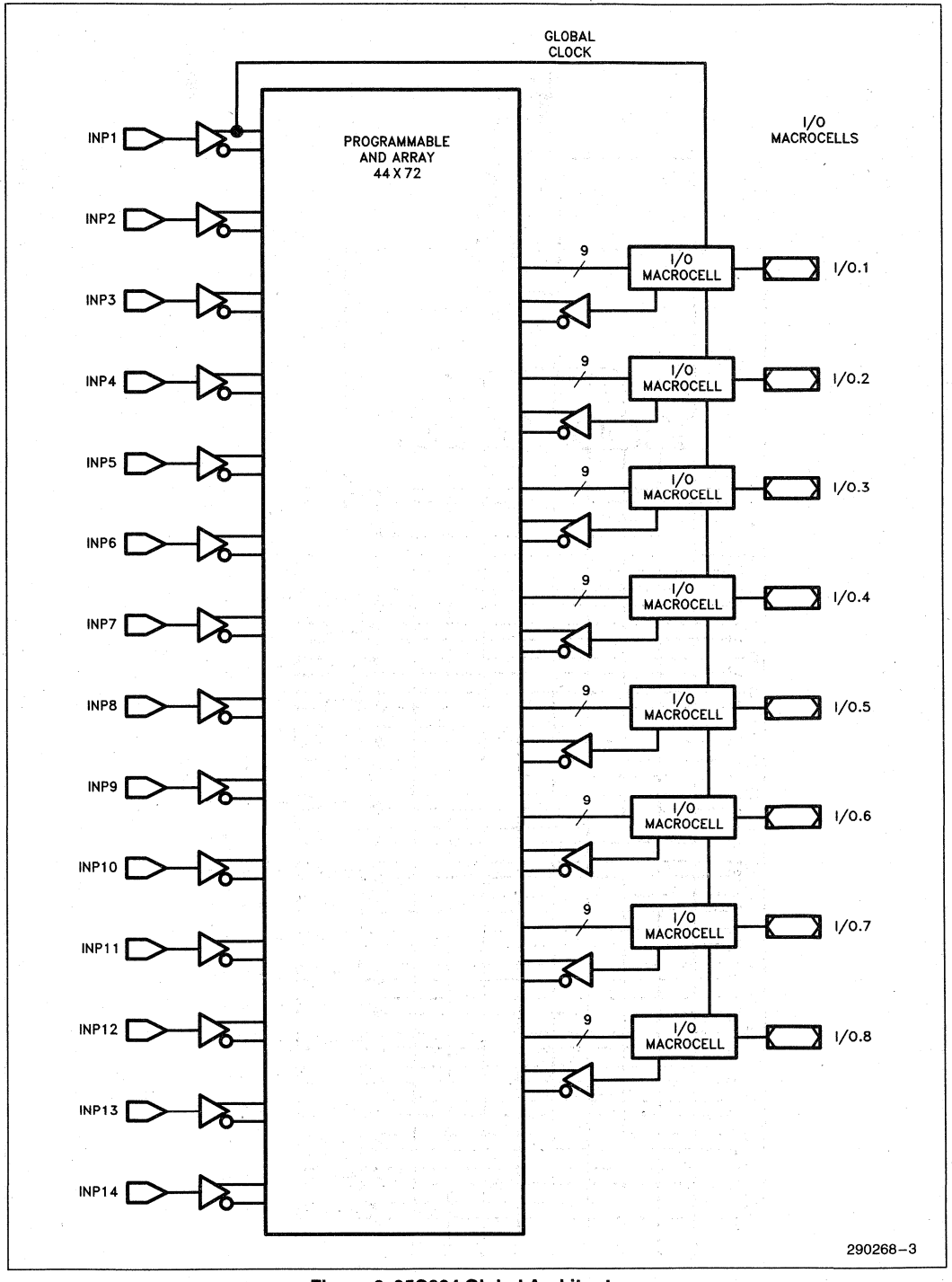


Figure 2. 85C224 Global Architecture

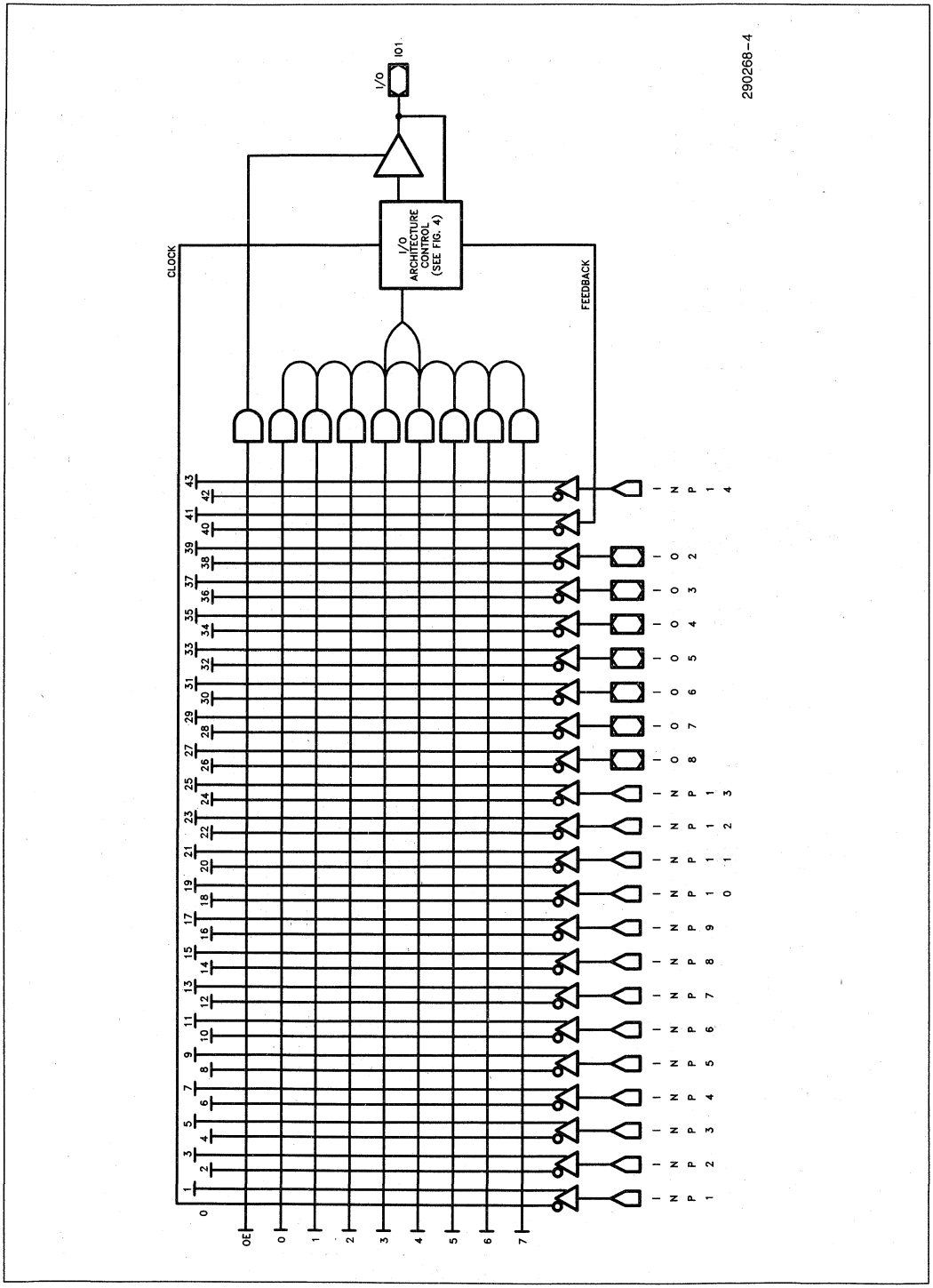


Figure 3. 85C224 Macrocell Architecture

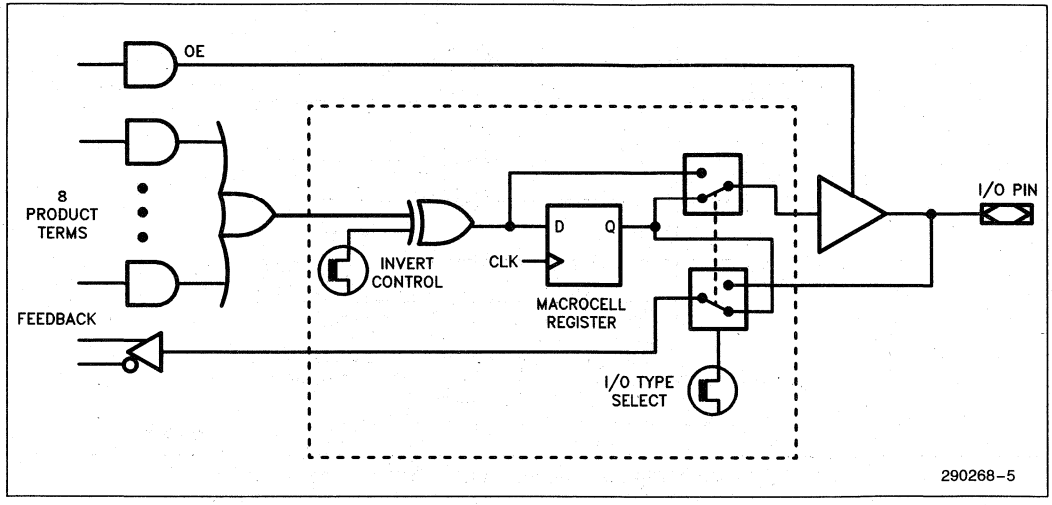


Figure 4. 85C224 I/O Control Architecture

AUTOMATIC STAND-BY MODE

The 85C224 contains a programmable bit, the Turbo Bit, that optimizes operation for speed or for power savings. When the Turbo Bit is programmed (TURBO = ON), the device is optimized for maximum speed. When the Turbo Bit is not programmed (TURBO = OFF), the device is optimized for power savings by entering standby mode during periods of inactivity.

Figure 5 shows the device entering standby mode approximately 75 ns after the last input or I/O transition. When the next input or I/O transition is detected, the device returns to active mode. Wakeup time adds an additional 20 ns to the propagation delay through the device as measured from the first transition. No delay occurs if an output is dependent on more than one input and the last of the inputs transitions after the device has returned to active mode. In addition, no delay occurs on any other output if the device has returned to active mode.

After erasure, the Turbo Bit is unprogrammed (OFF); automatic standby mode is enabled. When the Turbo Bit is programmed (ON), the device never enters standby mode.

POWER-ON CHARACTERISTICS

85C224 inputs and outputs begin responding 1 μs (max.) after V_{CC} power-up (V_{CC} = 4.75V) or after a power-loss/power-up sequence. All macrocells programmed as registers will be set to a logic low.

ERASED STATE CHARACTERISTICS

Prior to programming or after erasure, the I/O structure is configured for combinatorial, inverted output with input (pin) feedback.

ERASURE CHARACTERISTICS

Erasure time for the 85C224 is 1 hour at 12,000 μWsec/cm^2 with a 2537Å UV lamp.

Erasure characteristics of the device are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000Å. It should be noted that sunlight and certain types of flourescent lamps have wavelengths in the 3000Å–4000Å range. Data shows that constant exposure to room level flourescent lighting could erase

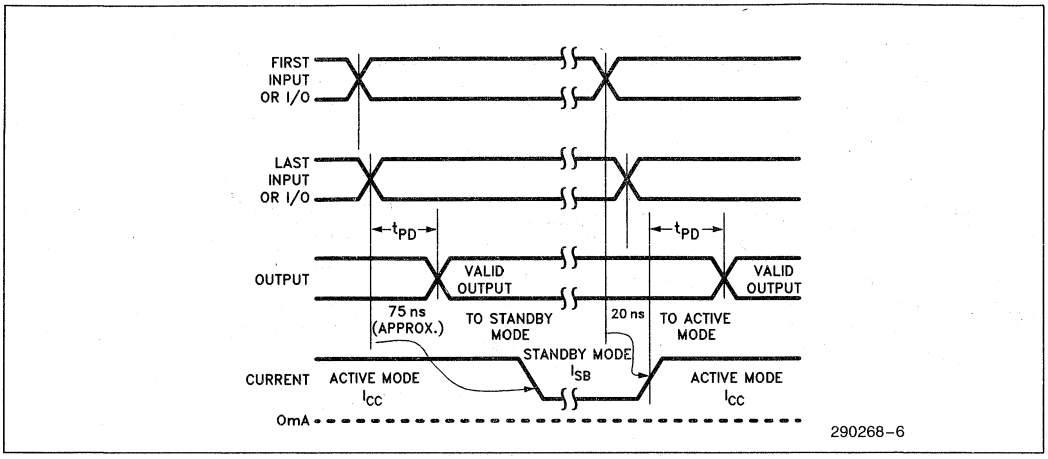


Figure 5. 85C224 Standby and Active Mode Transitions

the typical 85C224 in approximately six years, while it would take approximately two weeks to erase the device when exposed to direct sunlight. If the device is to be exposed to these lighting conditions for extended periods of time, conductive opaque labels should be placed over the device window to prevent unintentional erasure.

The recommended erasure procedure for the 85C224 is exposure to shortwave ultraviolet light with a wavelength of 2537Å. The integrated dose (i.e., UV intensity × exposure time) for erasure should be a minimum of forty (40) Wsec/cm^2. The erasure time with this dosage is one 1 hour using an ultraviolet lamp with a 12,000 μW/cm^2 power rating. The device should be placed within 1 inch of the lamp tubes during exposure. The maximum integrated dose the 85C224 can be exposed to without damage is 7258 Wsec/cm^2 (1 week at 12,000 μW/cm^2). Exposure to high intensity UV light for longer periods may cause permanent damage to the device.

inteligent Programming™ Algorithm

The 85C224 supports the inteligent Programming Algorithm, which rapidly programs Intel PLDs, while maintaining a high degree of reliability. It is particularly suited for production programming environments. This method ensures reliability as the incremental programming margin of each bit has been verified during programming. Programming voltage and waveform specifications are available from Intel by request to support device programming.

LATCH-UP IMMUNITY

All of the input, output, and clock pins of the device have been designed to resist latch-up which is inherent in inferior CMOS structures. The 85C224 is designed with Intel's proprietary 1-micron CHMOS IIIE

EPROM process. Thus, none of the pins will experience latch-up with currents up to ±100mA and voltages ranging from −0.5V to (V_{CC} + 0.5V). The programming pin is designed to resist latch-up to the 13.5V maximum device limit.

DESIGN RECOMMENDATIONS

For proper operation, it is recommended that all input and output pins be constrained to the voltage range GND ≤ (V_{IN} or V_{OUT}) ≤ V_{CC}. All unused inputs and I/Os should be tied high or low to minimize power consumption (do not leave them floating). A high-speed power supply decoupling capacitor of at least 0.1μF must be connected directly between the V_{CC} and GND pins.

As with all CMOS devices, ESD handling procedures should be used with the 85C224 to prevent damage to the device during programming, assembly, and test.

SOFTWARE SUPPORT

The 85C224 is supported by Version 2.2 or later of iPLS II (Intel Programmable Logic Software), which includes the LOC (Logic Optimizing Compiler) and APT (Advanced Programming Tool). Programming is supported by APT on the GUPI 24D28J Programming Adaptor using either an iUP-PC Personal Programmer or an iUP-200A/201A Universal Programmer.

For detailed information on iPLS II, refer to the iPLDS II Data Sheet, order number: 290134.

The 85C224 is also supported by third-party logic compilers such as ABEL*, CUPL*, PLDesigner*, Log/IC, etc. Programming support is provided by third-party programmer companies such as Data I/O, Logical Devices, STAG, etc. Please refer to the "Third-Party Support" lists in the *Programmable Logic* handbook for complete information and vendor contacts.

*ABEL is a trademark of Data I/O, Corp.
CUPL is a trademark of Logical Devices, Inc.
PLDesigner is a trademark of MINC, Inc.

ADF PRIMITIVES SUPPORTED

The following ADF primitives are supported by this device:

INP	RONF
CONF	RORF
COIF	NORF
NOCF	

2

ORDERING INFORMATION

f_{CNT1} (MHz)	f_{MAX} (MHz)	t_{PD} (ns)	Order Code	Package	Operating Range
80	111	10	D85C224-80	*CERDIP	Commercial
			P85C224-80	PDIP	
			N85C224-80	PLCC	
66	90.9	12	D85C224-66	*CERDIP	Commercial
			P85C224-66	PDIP	
			N85C224-66	PLCC	

*Only the windowed (CERDIP) package allows UV-erase.

ABSOLUTE MAXIMUM RATINGS*

Supply Voltage (V_{CC})(1) ..... −2.0V to +7.0V

Programming Supply Voltage (V_{PP})(1) ..... −2.0V to +13.5V

D.C. Input Voltage (V_I)(1,2) .... −0.5V to V_{CC} +0.5V

Storage Temperature (T_{stg}) ..... −65°C to +150°C

Ambient Temperature (T_{amb})(3) ... −10°C to +85°C

NOTES:
1. Voltages with respect to GND.
2. Minimum D.C. input is −0.5V. During transitions, the inputs may undershoot to −2.0V or overshoot to +7.0V for periods of less than 20 ns under no load conditions.
3. Under bias. Extended Temperature versions are also available.

**NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those listed in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability*

NOTICE: Specifications contained within the following tables are subject to change.

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Min	Max	Units
V_{CC}	Supply Voltage	4.75	5.25	V
V_{IN}	Input Voltage	0	V_{CC}	V
V_O	Output Voltage	0	V_{CC}	V
T_A	Operating Temperature	0	+70	°C
t_R	Input Rise Time		500	ns
t_F	Input Fall Time		500	ns

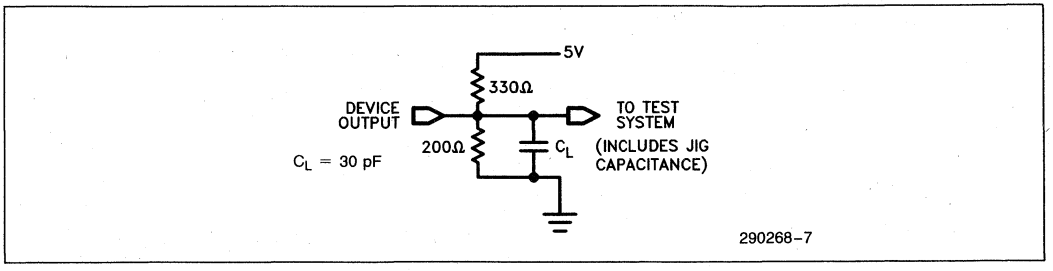
D.C. CHARACTERISTICS

(T_A = 0°C to +70°C, V_{CC} = 5.0V ±5%)

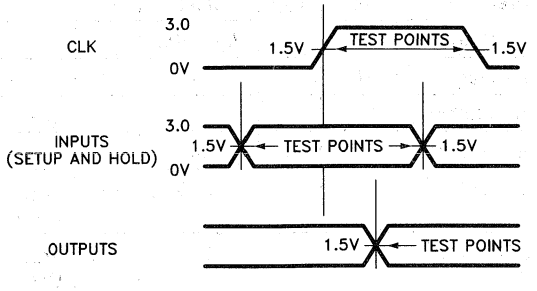
Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
V_{IH}(4)	High Level Input Voltage	2.0		V_{CC} +0.3	V	
V_{IL}(4)	Low Level Input Voltage	−0.3		0.8	V	
V_{OH}	High Level Output Voltage	2.4			V	I_O = −4.0 mA D.C., V_{CC} = min.
V_{OL}(5)	Low Level Output Voltage			0.45	V	I_O = 12.0 mA D.C., V_{CC} = min.
I_I	Input Leakage Current			±10	μA	V_{CC} = max., GND < V_{IN} < V_{CC}
I_{OZ}	Output Leakage Current			±10	μA	V_{CC} = max., GND < V_{OUT} < V_{CC}
I_{SC}(6)	Output Short Circuit Current	−30		−120	mA	V_{CC} = max., V_{OUT} = 0.5V
I_{SB}(7)	Standby Current		25	100	μA	V_{CC} = max., V_{IN} = V_{CC} or GND, Standby Mode
I_{CC}	Power Supply Current (see I_{CC} vs. Freq. graph)		2	5	mA	V_{CC} = max., V_{IN} = V_{CC} or GND, No Load, f_{IN} = 1 MHz, Device Prog. as an 8-Bit Counter, Non-Turbo Mode
			35	50	mA	f_{IN} = 15 MHz, Active Mode
			45	60	mA	f_{IN} = 80 MHz, Active Mode

NOTES:
4. Absolute values with respect to device GND; all over and undershoots due to system or tester noise are included.
5. Maximum DC I_{OL} for the device (all eight outputs) is 64 mA.
6. Not more than 1 output should be tested at a time. Duration of that test should not exceed 1 second.
7. In Non-Turbo Mode (TURBO=OFF), device enters standby mode approximately 75 ns after the last input transition.

A.C. TESTING LOAD CIRCUIT



A.C. TESTING WAVEFORM–SYNCHRONOUS INPUTS AND OUTPUTS



290268-8

A.C. Testing: Inputs (including CLK) are driven at 3.0V for a Logic "1" and OV for a Logic "0". Setup and hold input measurements relative to CLK are made at 1.5 V on CLK (low-to-high transition). Input and Output transitions are measured at 1.5V. Device input rise and fall times are less than 3 ns.

CAPACITANCE

(T_A = 0°C to +70°C; V_{CC} = 5.0V ±5%)[8]

Symbol	Parameter	Min	Typ	Max	Unit	Conditions
C_{IN}	Input Capacitance		4	6	pF	V_{IN} = 0V, f = 1.0 MHz
C_{IO}	I/O Capacitance		5	8	pF	V_{OUT} = 0V, f = 1.0 MHz
C_{CLK}	CLK Capacitance		6	8	pF	V_{IN} = 0V, f = 1.0 MHz
C_{VPP}	V_{PP} Pin Capacitance		8	10	pF	V_{PP} on INP12, f = 1.0 MHz

NOTES:
8. These values are evaluated during initial characterization and whenever design modifications occur that may affect characterization.

A.C. CHARACTERISTICS

(T_A = 0°C to +70°C, V_{CC} = 5.0V ±5%)[9]

Symbol	Parameter	85C224-80			85C224-66			Non-[10] Turbo	Units
		Min	Typ	Max	Min	Typ	Max		
t_{PD}[11]	Input or I/O to Output Valid	4		10	4		12	+20	ns
t_{PZX}[12]	Input or I/O to Output Enable	4		12	4		14	+20	ns
t_{PXZ}[12]	Input or I/O to Output Disable	4		10	4		12	+20	ns

SYNCHRONOUS CLOCK MODE

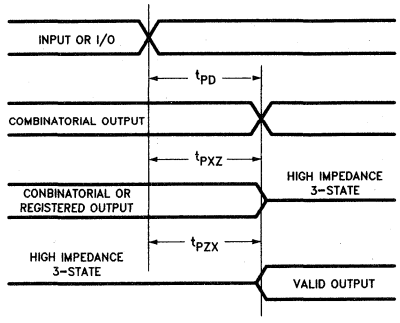
(T_A = 0°C to 70°C, V_{CC} = 5V ±5%)[9]

Symbol	Parameter	85C224-80			85C224-66			Non-[10] Turbo Mode	Units
		Min	Typ	Max	Min	Typ	Max		
f_{CNT1}[11]	Max. Counter Frequency $1/(t_{SU} + t_{CO})$ – Ext. Feedback		100	80		80	66		MHz
f_{CNT2}[11]	Max. Counter Frequency $1/(t_{CNT})$ – Internal Feedback		111	100		90	83.3		MHz
f_{MAX}[11]	Max. Frequency (Pipelined) $1/(t_{CW})$ – No Feedback		125	111		100	90.9		MHz
t_{SU}	Input or I/O Setup Time to CLK	7			9			+20	ns
t_H	Input or I/O Hold Time from CLK	0			0				ns
t_{CO1}	CLK High to Output Valid	1.5[13]		5.5[11]	1.5[13]		6[11]		ns
t_{CO2}	CLK High to Output Valid Fed Through Comb. Macrocell	4.5		13	4.5		15	+20	ns
t_{CNT}[11]	Macrocell Output Feedback to Macrocell Input – Internal Path	10			12			+20	ns
t_{CL}	CLK Low Time	4			5				ns
t_{CH}	CLK High Time	4			5				ns
t_{CW}	CLK Width	9			11				ns

NOTES:

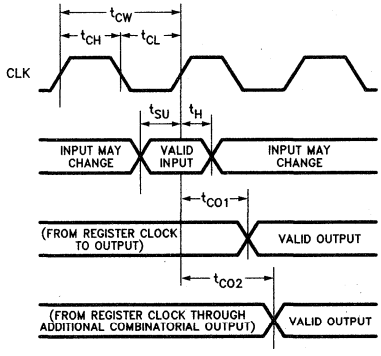
9. Typical values are at T_A = +25°C, V_{CC} = 5V, Active Mode.
10. If device is operated in Non-Turbo Mode (TURBO = OFF) and the device is inactive for approx. 75 ns, increase time by amount shown for power-up from standby mode.
11. Measured with all eight outputs switching.
12. t_{PZX} and t_{PXZ} are measured at ±0.5V from steady state voltage as driven by spec. output load. t_{PXZ} is measured with C_L = 5 pF. Measured with all eight outputs switching.
13. t_{CO1} min. is measured with one output switching, T_A = 0°C, V_{CC} = 5.25V.

COMBINATORIAL MODE



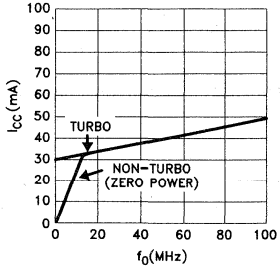
290268-9

REGISTERED MODE



290268-10

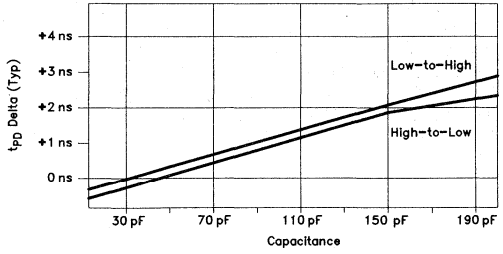
85C224
I_{CC} vs Frequency



Conditions:
T_A = 25°C
V_{CC} = 5.25V

290268-11

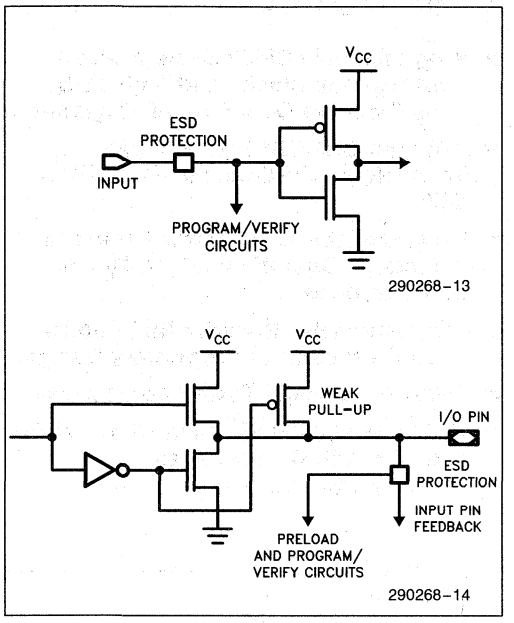
85C244
t_{PD} Derating vs Capacitive Loading



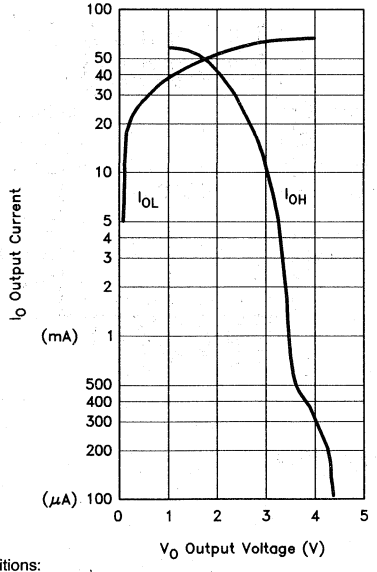
Conditions:
T_A = 70°C
V_{CC} = 4.75V

290268-12

INPUT/OUTPUT EQUIVALENT SCHEMATICS



85C224 Output Drive Current in Relation to Voltage



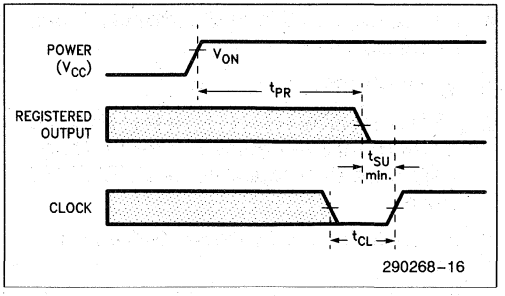
Conditions:
T_A = +80°C
V_{CC} = 4.75V

290268-15

Power-Up Reset

Internal power-up reset circuits ensure that all flip-flops will be reset to a logic 0 after the device has powered up. Because V_{CC} rise can vary significantly from one application to another, V_{CC} rise must be monotonic.

POWER-UP RESET



POWER-UP RESET CHARACTERISTICS

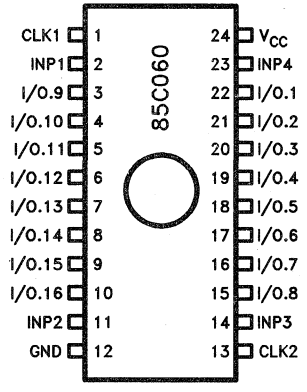
Symbol	Parameter	Value
t_{PR}	Power-Up Reset Time	1000 ns Max
V_{ON}	Turn-On Voltage	4.75V

2

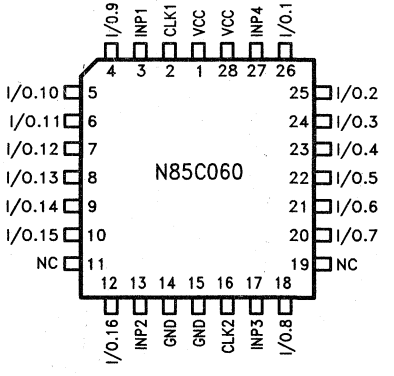
85C060
16-MACROCELL CHMOS μPLD

- High-Performance LSI Semi-Custom Logic Alternative to Low-End Gate Arrays, TTL, 74HC SSI, MSI Logic, and Bipolar PLDs
- High-Speed Upgrade to 5C060, EP600, EP610, EP630, and CE630
- t_{PD} 12 ns, 66 MHz w/Feedback, Clock to Output 7 ns
- I_{CC} = 80 mA @ 66 MHz
- 16 Macrocells with Programmable I/O Architecture (Register/Combinatorial). Registers Configurable as D/T/JK/RS Types
- Up to 20 Inputs (4 Dedicated and 16 I/O)
- 8 P-Terms, Selectable SOP Invert, Clear and OE P-Terms for Each Macrocell
- Programmable Clock System with 2 Synchronous Clocks and Asynchronous Clocking Option on all Registers
- 1-Micron CHMOS* III-E EPROM Technology. UV-Erasable (CerDIP) or OTP
- Programmable Low-Power Option for "Standby" Operation; 20 μA Typ. in Standby Mode
- Programmable "Security Bit" Allows Total Protection of Proprietary Designs
- 100% Generically Tested Logic Array
- Available in 300-mil 24-Pin CerDIP/PDIP and 28 Pin PLCC Packages

(See Packaging Spec. Order #231369)



290246-1



290246-2

Figure 1. 85C060 Pin Configurations

*CHMOS is a patented process of Intel Corporation.

The 85C060 is a high-performance, high-integration, general-purpose CMOS PLD. The 85C060 μPLD (Microcomputer Programmable Logic Device) accommodates logic functions with up to 20 inputs and 16 I/O macrocells. Each I/O macrocell includes 8 p-terms for input, a separate clear p-term, and an output enable/asynchronous clock p-term. With a maximum external feedback frequency of 66 MHz, the 85C060 is well suited to high-performance microprocessor-based systems.

The 85C060 uses CHMOS EPROM (floating gate) cells as logic control elements instead of fuses. The CHMOS EPROM technology reduces power consumption in comparison to bipolar devices without sacrificing speed performance. In addition, Intel's advanced CHMOS III-E EPROM process technology enables higher logic densities to be achieved with superior speed and low-power performance over other comparable devices. Intel's μPLDs add the benefits of "zero" stand-by power not available on other programmable logic devices. EPROM technology allows these devices to be 100% factory tested by programming and erasing all the EPROM logic control elements.

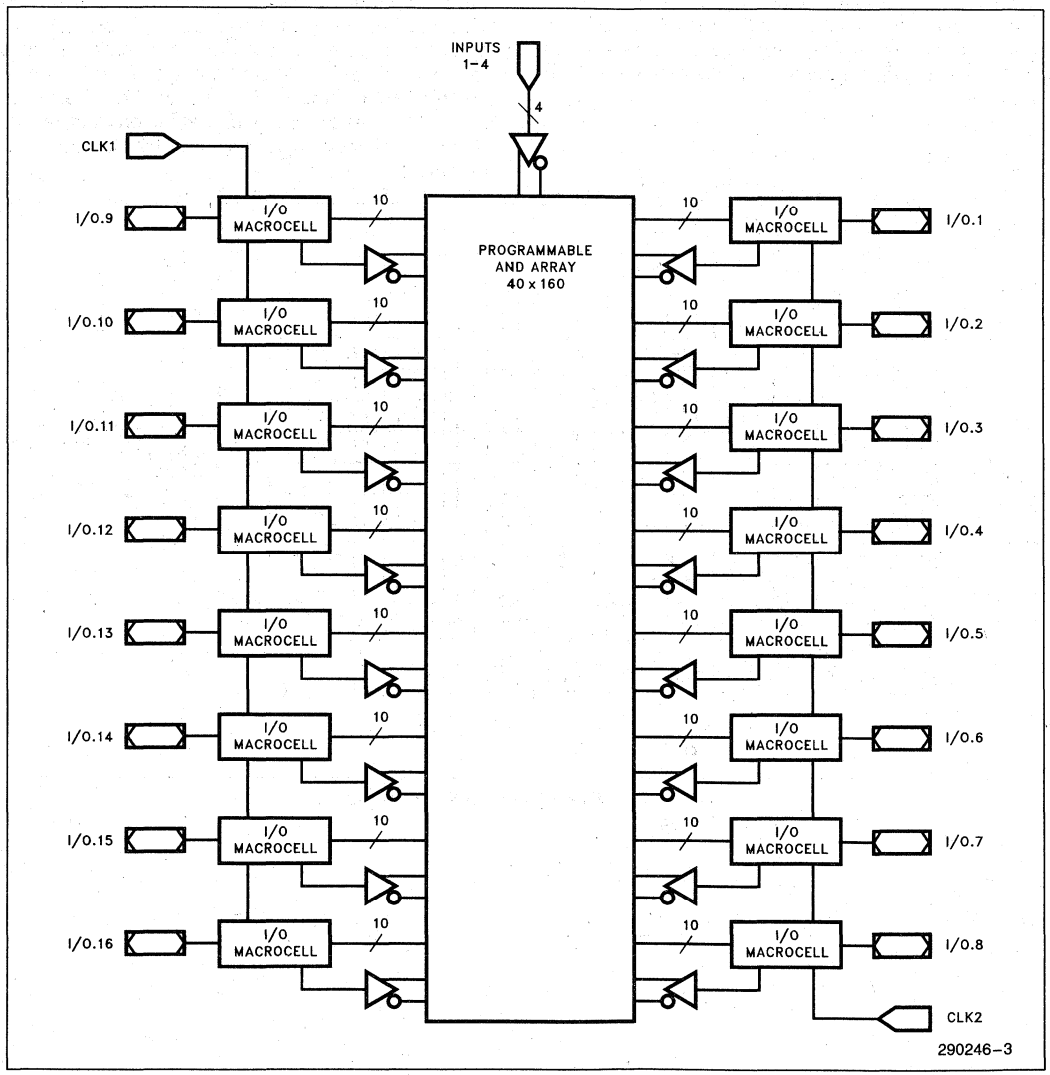


Figure 2. 85C060 Global Architecture

The architecture of the 85C060 is based on the "Sum of Products" PLA (Programmable Logic Array) structure with a programmable AND array feeding into a fixed OR array. The device accommodates combinational and sequential logic functions. A programmable I/O architecture provides individual selection of either combinatorial or registered output and feedback signals all with selectable polarity.

A feature unique to the 85C060 is the ability to individually program the output registers as a D-, T-, SR-, or JK-type Flip-Flop without sacrificing the utilization of programmable AND logic. Each output register can be individually clocked from any of the input or feedback paths available within the AND array. With these features, a wide variety of logic functions can be simultaneously implemented–all on the same device.

ARCHITECTURE DESCRIPTION

Externally, the 85C060 has 4 dedicated data input pins, 16 I/O pins that may be configured for input, output, or bidirectional operations, and 2 synchronous clock inputs. The 85C060 is contained in a 24-pin ceramic windowed or plastic package (300 mils) or 28-lead J-leaded chip carrier package.

The basic Macrocell architecture for the 85C060 is shown in Figure 3. The 85C060 has 16 of these Macrocells (one for each I/O pin). The Macrocell is organized in the familiar sum-of-products structure with a programmable AND array attached to a fixed OR term. The inputs to the programmable AND array originate from the true and complement signals from each of the dedicated input pins and each of the I/O control blocks. The 40-input AND array of the 85C060 feeds 160 AND gates (product terms) which are distributed among the 16 Macrocells in the device.

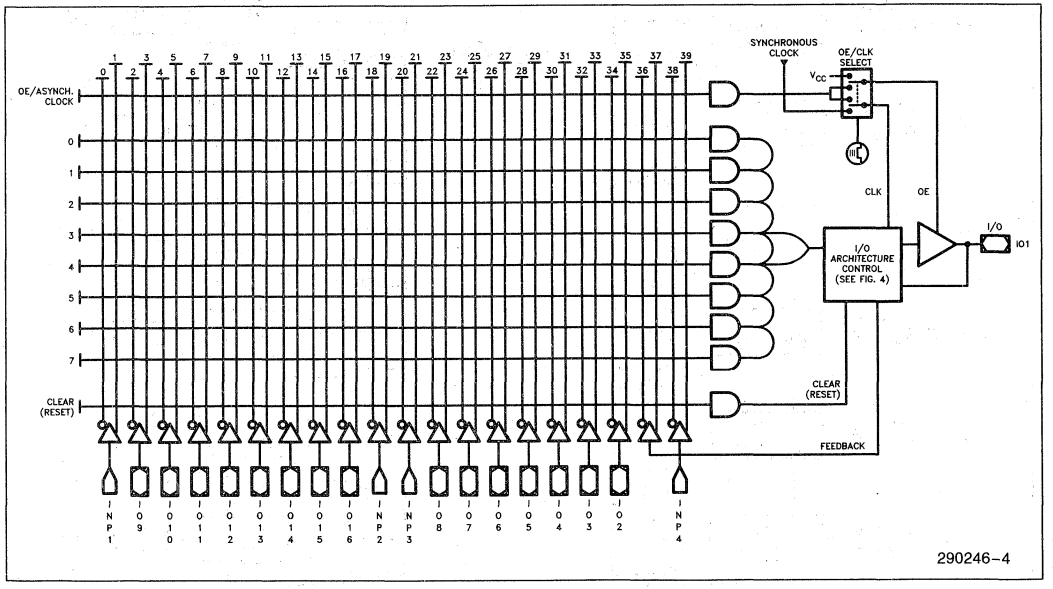


Figure 3. 85C060 Macrocell Architecture

Each Macrocell contains ten product terms. Eight of the ten product terms (AND gates) are dedicated for the SOP logic implementation. One product term on each Macrocell is used for RESET control to the output register associated with the Macrocell. The final product term is used for OE/Asynchronous Clock implementation.

Within the AND array, there is an EPROM connection at every intersection of an input signal (true and complement) and a product term to a given Macrocell. Before programming an erased device, every EPROM connection is made at every intersection. But during the programming process, these connections are opened so that only the desired connections remain. Therefore, the true or complement of any input signal can be connected to any product term. If both the true and complement connections of any signal are left intact, a logical false results on the output of the AND gate. However, if both the true and complement connections are open, then a logic "don't care" results on the AND gate. Lastly, if all the inputs of a product term are programmed open, then a logical true results on the output of the AND gate.

The 85C060 has two dedicated clock inputs to provide synchronous clock signals to the internal registers. Each of the clock signals controls half the total registers within the given device. For example, CLK1 provides synchronous clocking to the registers in Macrocells in the left half of the array while CLK2 controls the registers associated with Macrocells in the right half of the array. The advanced I/O architecture allows for any number of the registers to be synchronously clocked (from none to all). Both of the dedicated clock inputs latch the data into a given register when triggered on a positive edge.

MACROCELL ARCHITECTURE SELECTION

The 85C060 architecture provides each Macrocell with over 50 different possible I/O register configurations. Each I/O pin can be configured for combinatorial or registered output (true or complement) with feedback. In addition, four different types of output registers can be implemented on I/O pin without any additional logic requirements. The feedback mechanism for each register back into the AND array can be programmed to provide for either registered feedback from the Macrocell or input feedback (treating the pin as an input). Another advantage of the advanced I/O capability of the 85C060 is the ability to individually clock each internal register from asynchronous clock signals.

Invert Select EPROM Bit

The Invert Select EPROM bit is used to invert the product term input into the register. This applies to all inputs including double inputs on the JK and SR registers.

REGISTER SELECTION

The advanced I/O architecture of the 85C060 allows four different register types along with combinatorial output as illustrated in Figure 4a. The register types include a T, D, JK, or SR Flip-Flop and each Macrocell I/O structure may be independently configured. In addition, all registers have an individual asynchronous RESET control from a dedicated product term derived in the AND array. When this dedicated product term is a logical one, the Macrocell register is immediately cleared to a logical zero independent of the register clock. The RESET function occurs automatically on power-up.

Output Register Configuration

The four different register types shown in Figure 4b–4e are described below.

D- or T-type Flip-Flops

When either a D- or T-type Flip-Flop is configured as part of the I/O structure, all eight of the product terms into the Macrocell are ORed together and fed into the register input.

JK or SR Registers

When either a JK or SR register is configured, the eight product terms are shared among two OR gates (one for the J or S input and the other for the K or R input). The allocation for these product terms for each of the register inputs is optimized by the iPLS II software.

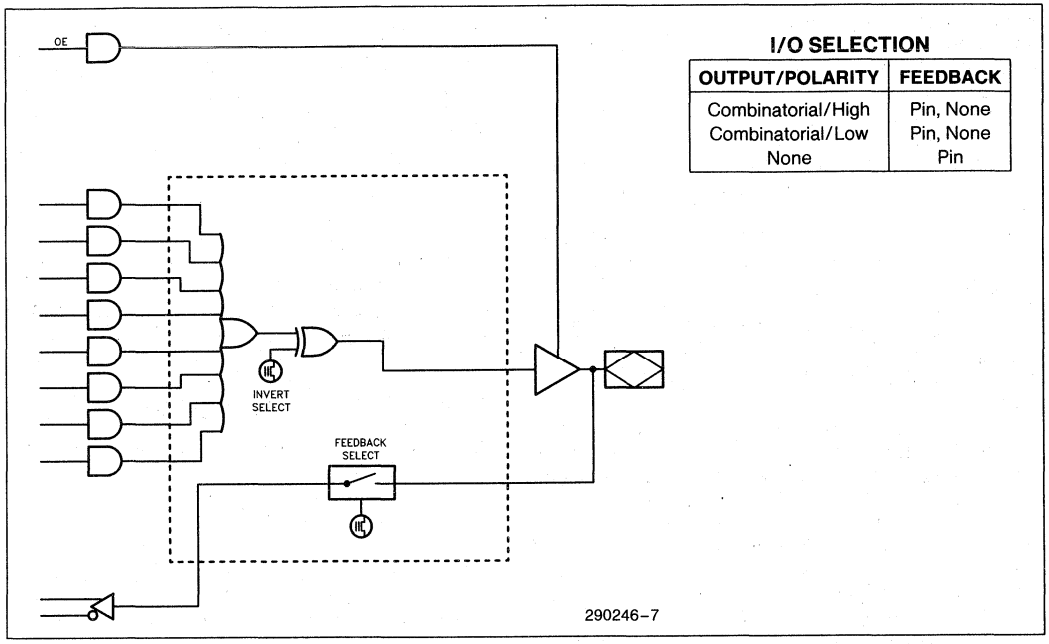
OUTPUT/FEEDBACK

The Output Select Multiplexer allows for either registered, combinatorial or no output.

The Feedback Select Multiplexer EPROM bit enables registered, I/O (using the pin for bidirectional input or just input), or no feedback to the AND array.

The Feedback Select is also important for building product terms with more than 8 products. The 8-product product term of a Macrocell can be fed back into the AND array and combined with still more signals to create a much larger product term (of more than 8-inputs). In addition, if the feedback product term is not to be output, then the iPLS II will reserve the associated Macrocell pin and indicate it in the REPORT file. A reserved pin should be left floating (no connect) when assembled onto a circuit board.

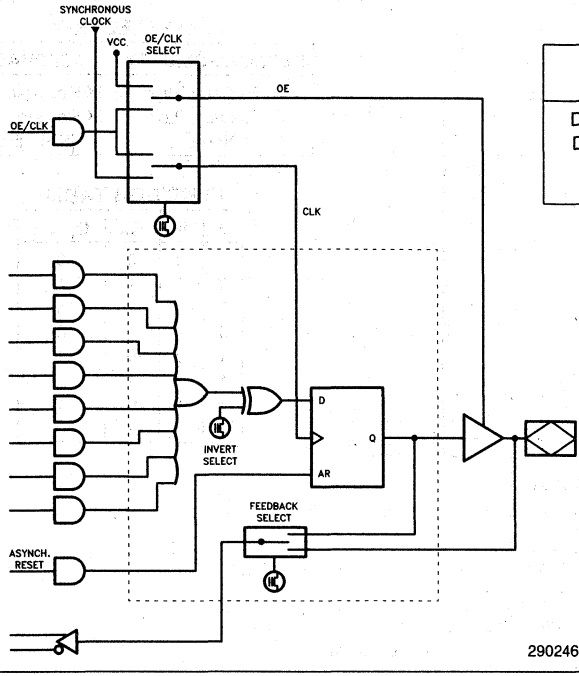
Any I/O pin may be configured as a dedicated input by selecting no output and pin feedback through the appropriate multiplexers.



I/O SELECTION

OUTPUT/POLARITY	FEEDBACK
Combinatorial/High	Pin, None
Combinatorial/Low	Pin, None
None	Pin

Figure 4a. Combinatorial I/O Configuration



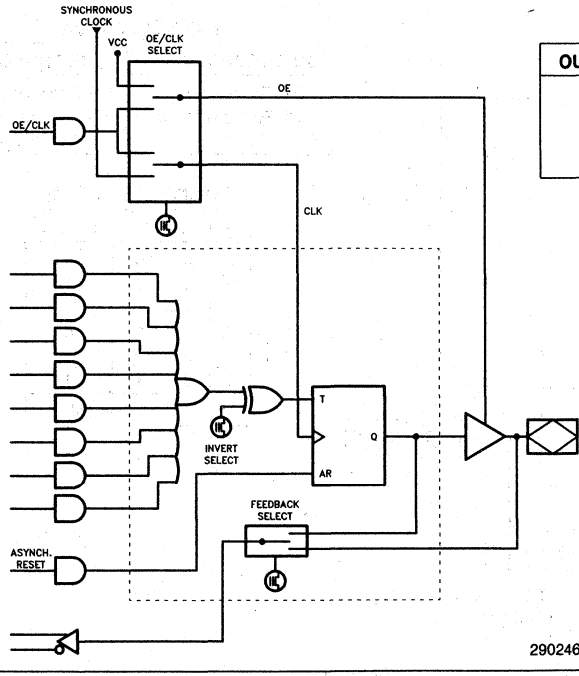
I/O SELECTION

OUTPUT/ POLARITY	FEEDBACK
D-Register/High	D-Register, Pin, None
D-Register/Low	D-Register, Pin, None
None	D-Registered
None	Pin

FUNCTION TABLE

D	Q_n	$Q_n + 1$
0	0	0
0	1	0
1	0	1
1	1	1

Figure 4b. D-Type Flip-Flop Register Configuration



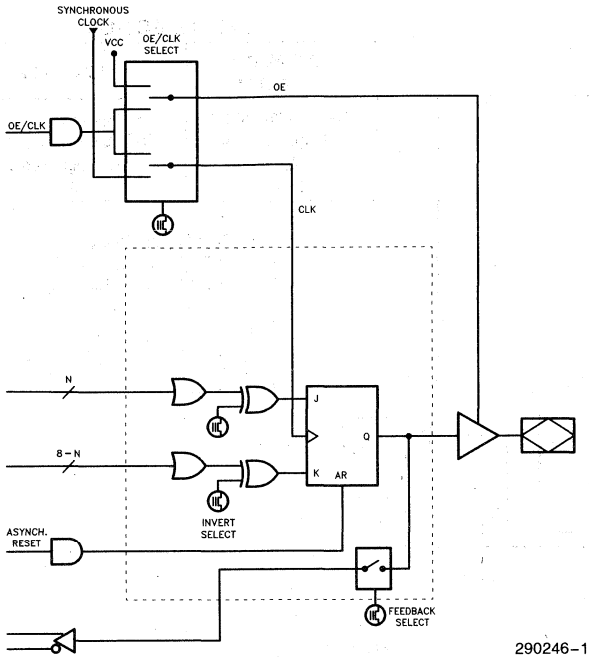
I/O SELECTION

OUTPUT/POLARITY	FEEDBACK
T-Register/High	T-Register, Pin, None
T-Register/Low	T-Register, Pin, None
None	T-Register
None	Pin

FUNCTION TABLE

T	Q_n	$Q_n + 1$
0	0	0
0	1	1
1	0	1
1	1	0

Figure 4c. Toggle Flip-Flop Register Configuration



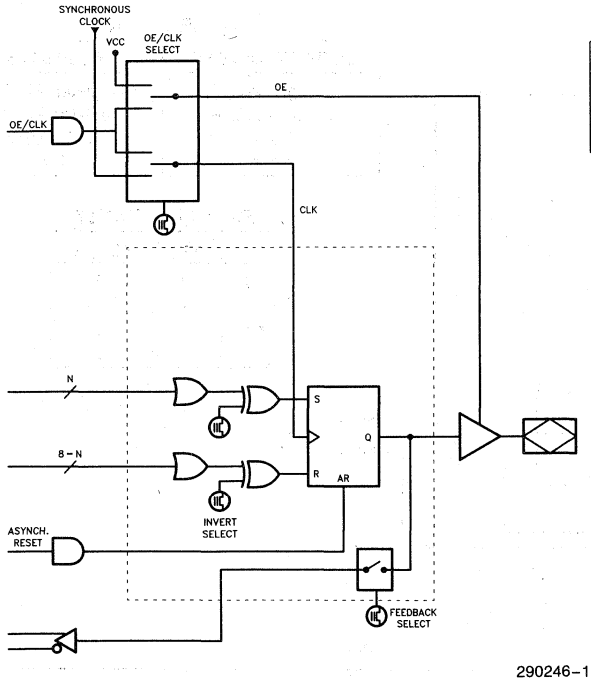
I/O SELECTION

OUTPUT/POLARITY	FEEDBACK
JK Register/High	JK Register, None
JK Register/Low	JK Register, None
None	JK Register

FUNCTION TABLE

J	K	Q_n	$Q_n + 1$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Figure 4d. JK Flip-Flop Register Configuration



I/O SELECTION

OUTPUT/POLARITY	FEEDBACK
SR Register/High	SR Register, None
SR Register/Low	SR Register, None
None	SR Register

FUNCTION TABLE

S	R	Q_n	$Q_n + 1$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	Illegal	

Figure 4e. SR Flip-Flop Register Configuration

Output Enable (OE)/Clock Selection

Two modes of operation are provided by the OE/CLK Select Multiplexer as a part of each Macrocell. One mode provides for three-state buffering of outputs while in the other mode, the outputs are always enabled. The operation of the OE/CLK Select Multiplexer sets the mode within a given Macrocell. Therefore, the output mode can be selected individually on every output. Figure 5 illustrates the two modes of OE/CLK operation.

MODE 0: THREE-STATE BUFFERING

In Mode 0, the three-state output buffer is controlled by a single product term originating from the AND array. The output is enabled when the product term is a logical true. Conversely, the output appears as high impedance when the product term is a logical false as shown in Table 1. In Mode 0, the Macrocell Flip-Flop is connected to its associated synchronous clock (either CLK1 or CLK2 depending upon the Macrocell's location within the device). Thus, the Macrocell Flip-Flop may be clocked by its respective synchronous clock but its output will not become valid until the output is enabled.

Table 1. Mode 0 Output Selection

Product Term	Output Buffer
FALSE	Three-State
TRUE	Enabled

MODE 1: OUTPUT BUFFER ENABLED

In Mode 1, the Output Buffer is always enabled. In addition, the Macrocell Flip-Flop is connected to the AND array. The Macrocell Flip-Flop may now be triggered from an asynchronous clock signal generated by the AND array logic to the OE/CLK multiplexable term. Mode 1 allows the Macrocell Flip-Flops to be individually clocked from any of the available signals in the AND array. Since both true and complement values appear in the AND array, the Flip-Flop may be clocked by any positive- or negative-going signals at any input pin. Gated clock structures can be created since the Flip-Flop clock is created by a product term.

AUTOMATIC STAND-BY MODE

The 85C060 contains a programmable bit, the Turbo Bit, that optimizes operation for speed or for power savings. When the Turbo Bit is programmed (TURBO = ON), the device is optimized for maximum speed. When the Turbo Bit is not programmed (TURBO = OFF), the device is optimized for power savings by entering standby mode during periods of inactivity.

Figure 6 shows the device entering standby mode approximately 100 ns after the last input transition. When the next input transition is detected, the device returns to active mode. Wakeup time adds an additional 25 ns to the propagation delay through the device as measured from the first input. No delay will occur if an output is dependent on more than one input and the last of the inputs changes after the device has returned to active mode.

After erasure, the Turbo Bit is unprogrammed (OFF); automatic standby mode is enabled. When the Turbo Bit is programmed (ON), the device never enters standby mode.

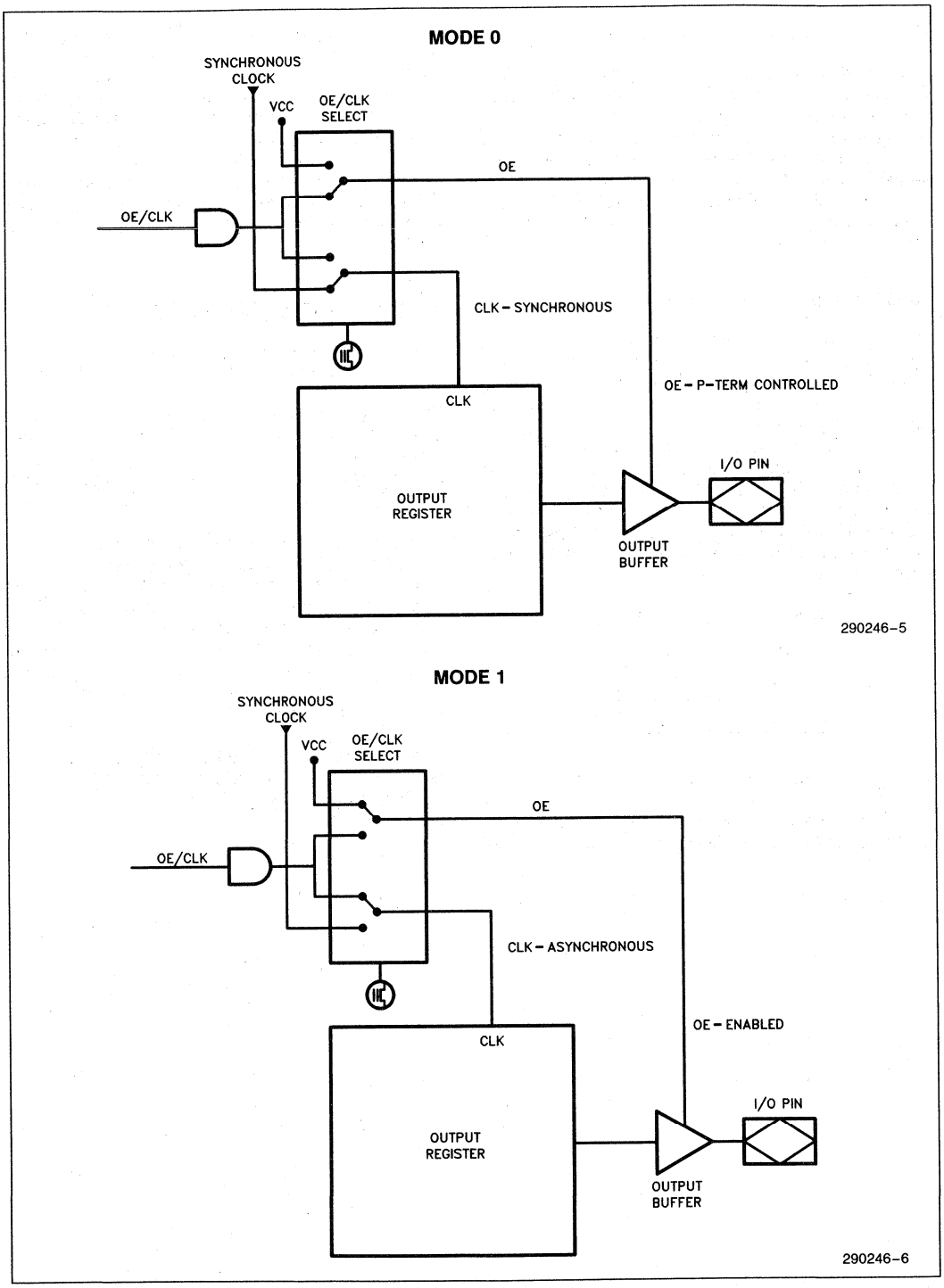


Figure 5. Output Enable/Clock Configuration

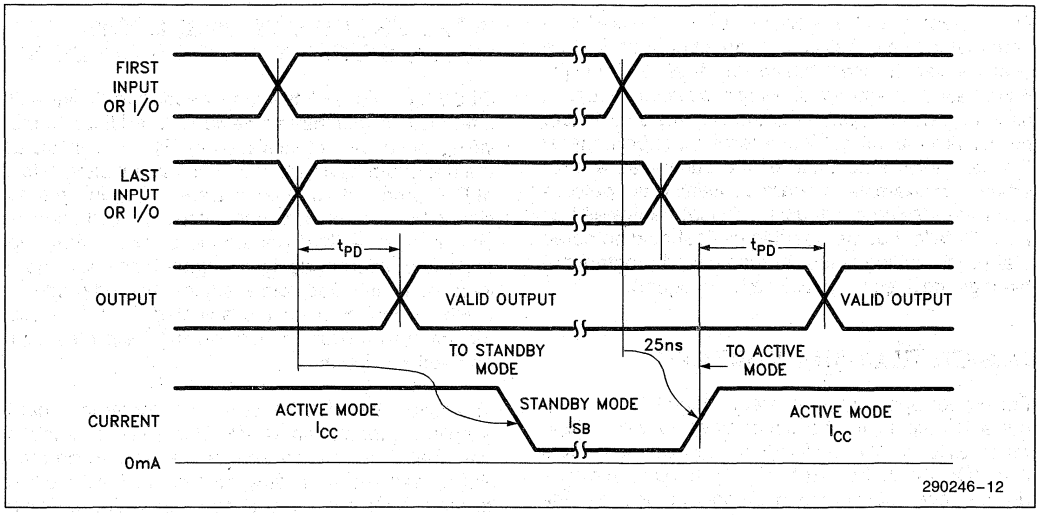


Figure 6. 85C060 Standby and Active Mode Transitions

Erased-State Configuration

Prior to programming or after erasing, the I/O structure is configured for combinatorial active low output with input (pin) feedback.

ERASURE CHARACTERISTICS

Erase time for the 85C060 is 1 hour at 12,000 $\mu W/cm^2$ with a 2537Å lamp.

Erasure characteristics of the device are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000Å. It should be noted that sunlight and certain types of flourescent lamps have wavelengths in the 3000Å–4000Å. Data shows that constant exposure to room level flourescent lighting could erase the typical device in approximately six years, while it would take approximately two weeks to cause erasure when exposed to direct sunlight. If the 85C060 is to be exposed to these types of lighting conditions for extended periods of time, conductive opaque labels should be placed over the device window to prevent unintentional erasure.

The recommended erasure procedure for the 85C060 is exposure to shortwave ultraviolet light with a wavelength of 2537Å. The integrated dose (i.e., UV intensity × exposure time) for erasure should be a minimum of forty (40) $Wsec/cm^2$. The erasure time with this dosage is approximately 1 hour using an ultraviolet lamp with a 12,000 $\mu W/cm^2$ power rating. The 85C060 should be placed within one inch of the lamp tubes during erasure.

The maximum integrated dose the 85C060 can be exposed to without damage is 7258 $Wsec/cm^2$ (1 week at 12,000 $\mu W/cm^2$). Exposure to high intensity UV light for longer periods may cause permanent damage to the device.

PROGRAMMING CHARACTERISTICS

Initially, and after erasure, all the EPROM control bits of the 85C060 are connected (in the "1" state). Each of the connected control bits are selectively disconnected by programming the EPROM cells into their "0" state. Programming voltage and waveform specifications are available by request from Intel to support programming of the 85C060.

int$_e$ligent Programming™ Algorithm

The 85C060 supports the int$_e$ligent Programming Algorithm which rapidly programs Intel PLDs using an efficient and reliable method. This method ensures reliability as the incremental program margin of each bit is continually monitored to determine when the bit has been successfully programmed.

FUNCTIONAL TESTING

Since the logical operation of the 85C060 is controlled by EPROM elements, the device is completely testable. Each programmable EPROM bit controlling the internal logic is tested using application-independent test program patterns. After testing, the devices are erased before shipment to customers. No post-programming tests of the EPROM array are required.

The testability and reliability of EPROM-based programmable logic devices is an important feature over similar devices based on fuse technology. Fuse-based programmable logic devices require a user to perform post-programming tests to insure proper programming. These tests must be done at the device level because of the cummulative error effect. For example, a board containing ten devices each possessing a 2% device fallout translates into an 18% fallout at the board level (it should be noted that programming fallout of fuse-based programmable logic devices is typically 2% or higher).

DESIGN RECOMMENDATIONS

For proper operation, it is recommended that all input and output pins be constrained to the voltage range GND $<$ (V_{IN} or V_{OUT}) $<$ V_{CC}. Unused inputs and I/Os should be tied to V_{CC} or GND to minimize device power consumption. Reserved pins (as indicated in the logic compiler REPORT file) should be left floating (no connect) so that the pin can attain the appropriate logic level. A power supply decoupling capacitor of at least 0.2 μF must be connected directly between V_{CC} and GND pins of the device.

As with all CMOS devices, ESD handling procedures should be used with the 85C060 to prevent damage to the device during programming, assembly, and test.

DESIGN SECURITY

A single EPROM bit provides a programmable design security feature that controls the access to the data programmed into the device. If this bit is set, a proprietary design within the device cannot be copied. This EPROM security bit enables a higher degree of design security than fused-based devices since programmed data within EPROM cells is invisible even to microscopic evaluation. The EPROM security bit, along with all the other EPROM control bits, will be reset by erasing the device.

LATCH-UP IMMUNITY

All of the input, I/O, and clock pins of the 85C060 have been designed to resist latch-up which is inherent in inferior CMOS structures. The 85C060 is designed with Intel's proprietary CHMOS II-E EPROM process. Thus, each of the pins will not experience latch-up with currents up to $\pm$100 mA and voltages ranging from $-$1V to (V_{CC} + 1V). Furthermore, the programming pin is designed to resist latch-up to the 13.5V maximum device limit.

INTEL PROGRAMMABLE LOGIC DEVELOPMENT SYSTEM II (iPLDS II)

iPLDS II provides all the tools needed to design with Intel PLDs or compatible devices. In addition to providing development assistance, iPLDS II insulates the user from having to know all the intricate details of PLD architecture (the software will optimize a design to benefit from architectual features). It contains comprehensive third generation software that supports four different design entry methods, minimizes logic, does automatic pin assignments and produces the best design fit for the selected PLD. It is user friendly with guided menus, on-line Help messages and soft key inputs.

In addition, the iPLDS II contains programmer hardware in the form of an iUP-PC Universal Programmer Personal Computer to enable the user to program PLDs, read and verify programmed devices and also to graphically edit programming files. The software generates industry standard JEDEC object code output files which can be downloaded to other programmers as well.

iPLS II software interfaces to several schematic capture packages to enable designs to be entered in schematic form. IPLDview-286/IPLDdraw allows the designer to use familiar TTL symbols or EPLD design primitive symbols. User-defined symbols are also supported. IPLDdraw provides a path to A.C. timing simulation of EPLD designs.

SCHEMA III-PLD allows the designer to use TTL symbols, EPLD custom macros, or EPLD design primitive symbols. It also supports user-defined symbols.

Other design formats include Boolean equation entry (supported directly by iPLS II) and state machine entry (supported by iSTATE).

Detailed information on the Intel Programmable Logic Development System II is contained in a separate Intel data sheet. (Order Number: 280168). Refer to the tools section of the *Programmable Logic* handbook for a complete listing of development tools.

The 85C060 is also supported by third-party logic compilers such as ABEL*, CUPL*, PLDesigner*, Log/IC, etc. Programming support is provided by third-party programmer companies such as Data I/O, Logical Devices, STAG, etc. Please refer to the "Third-Party Support" lists in the *Programmable Logic* handbook for complete information and vendor contacts.

*ABEL is a trademark of Data I/O Corp. CUPL is a trademark of Logical Devices, Inc. PLDesigner is a trademark of MINC, Inc.

ADF PRIMITIVES SUPPORTED

The following ADF primitives are supported by this device:

INP	JOJF
CONF	JONF
COIF	SONF
RONF	SOSF
RORF	TOIF
ROIF	TONF
NORF	TOTF
NOJF	CLKB
NOSF	
NOTF	

ORDERING INFORMATION

f_{CNT1} (MHz)	f_{MAX} (MHz)	t_{PD} (ns)	Order Code	Package	Operating Range
66	100	12	D85C060-12	*CerDIP	Commercial
			P85C060-12	PDIP	
			N85C060-12	PLCC	
50	83.3	15	D85C060-15	*CerDIP	Commercial
			P85C060-15	PDIP	
			N85C060-15	PLCC	
40	66	25	D85C060-25	*CerDIP	Commercial
			P85C060-25	PDIP	
			N85C060-25	PLCC	

*Windowed CerDIP package allows UV erase.

2

ABSOLUTE MAXIMUM RATINGS*

Symbol	Parameter	Min	Max	Units
V_{CC}	Supply Voltage(1)	−2.0	7.0	V
V_{PP}	Programming Supply Voltage(1)	−2.0	13.5	V
V_I	DC Input Voltage(1)(2)	−0.5	V_{CC} + 0.5	V
t_{stg}	Storage Temperature	−65	+150	°C
t_{amb}	Ambient Temperature(3)	−10	+85	°C

NOTICE: This data sheet contains preliminary information on new products in production. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

NOTES:
1. Voltages with respect to ground.
2. Minimum DC input is −0.5V. During transitions, the inputs may undershoot to −2.0V or overshoot to +7.0V for periods less than 20 ns under no load conditions.
3. Under bias. Extended temperature versions are also available.

RECOMMENDED OPERATING CONDITIONS

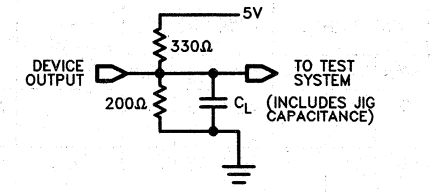
Symbol	Parameter	Min	Max	Unit
V_{CC}	Supply Voltage	4.75	5.25	V
V_{IN}	Input Voltage	0	V_{CC}	V
V_O	Output Voltage	0	V_{CC}	V
T_A	Operating Temperature	0	+70	°C
t_R	Input Rise Time		500	ns
t_F	Input Fall Time		500	ns

D.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5.0V ±5%

Symbol	Parameter	Min	Typ	Max	Unit	Conditions
V_{IH}(4)	HIGH Level Input Voltage	2.0		V_{CC} + 0.3	V	
V_{IL}(4)	LOW Level Input Voltage	−0.3		0.8	V	
V_{OH}	HIGH Level Output Voltage	2.4			V	I_O = −4.0 mA DC, V_{CC} = Min.
V_{OL}(5)	LOW Level Output Voltage			0.45	V	I_O = 12.0 mA DC, V_{CC} = Min.
I_I	Input Leakage Current	−10		10	µA	V_{CC} = Max., GND < V_{IN} < V_{CC}
I_{OZ}	Output Leakage Current	−10		10	µA	V_{CC} = Max., GND < V_{OUT} < V_{CC}
I_{SC}(6)	Output Short Circuit Current	−30		−120	mA	V_{CC} = Max., V_{OUT} = 0.5V
I_{SB}(7)	Standby Current		20	100	µA	V_{CC} = Max., V_{IN} = V_{CC} or GND, Standby Mode
I_{CC}	Power Supply Current (See I_{CC} vs. Freq. Graph)		3	8	mA	V_{CC} = Max, V_{IN} = V_{CC} or GND, No Load, f_{IN} = 1 MHz, Device Prog. as 16-Bit Counter, Turbo = Off
			60	80	mA	Turbo = On, f_{IN} = 66 MHz

NOTES:
4. Absolute values with respect to device GND; all over and undershoots due to system or tester noise are included.
5. Maximum DC I_{OL} for the device is 64 mA for CLK1 group I/O. 1–I/O.8 and 64 mA for CLK2 group I/O.9–I/O16.
6. Not more than 1 output should be tested at a time. Duration of that test must not exceed 1 second.
7. In Non-Turbo Mode (TURBO = OFF), device enters standby mode approximately 75 ns after the last input transition. I_{SB} is measured with the window covered (CerDIP).

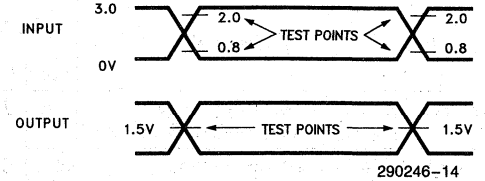
A.C. TESTING LOAD CIRCUIT



290246-13

C_L = 30 pF

A.C. TESTING INPUT, OUTPUT WAVEFORM



290246-14

A.C. Testing: Inputs are Driven at 3.0V for a Logic "1" and 0V for a Logic "0". Timing Measurements are made at 2.0V for a Logic "1" and 0.8V for a Logic "0" on inputs. Outputs are measured at a 1.5V point. Device input rise and fall times are less than 3 ns.

CAPACITANCE (T_A = 0°C to +70°C; V_{CC} = 5.0V ±5%)(8)

Symbol	Parameter	Min	Typ	Max	Unit	Conditions
C_{IN}	Input Capacitance		5	8	pF	V_{IN} = 0V, f = 1.0 MHz
C_{IO}	I/OCapacitance		6	8	pF	V_{OUT} = 0V, f = 1.0 MHz
C_{CLK}	CLK Capacitance		8	10	pF	V_{IN} = 0V, f = 1.0 MHz
C_{VPP}	V_{PP} Pin Capacitance		10	12	pF	V_{PP} on CLK2, f = 1.0 MHz

NOTE:
8. These values are evaluated during initial characterization and whenever design modifications occur that may affect capacitance.

COMBINATORIAL MODE A.C. CHARACTERISTICS

(T_A = 0°C to +70°C, V_{CC} = 5.0V ±5%)(9)

Symbol	Parameter	85C060-12			85C060-15			85C060-25			Non-Turbo(10) Mode	Units
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
t_{PD1}(11)	Input to Output Valid			12			15			25	+25	ns
t_{PD2}(11)	I/O to Output Valid			12			15			25	+25	ns
t_{PZX}(12)	Input or I/O to Output Enable			15			18			25	+25	ns
t_{PXZ}(12)	Input or I/O to Output Disable			15			18			25	+25	ns
t_{CLR}	Input or I/O to Asynch. Reset			15			18			25	+25	ns

NOTES:
9. Typical values are at T_A = 25°C, V_{CC} = 5V, Active Mode.
10. If device is operated in Non-Turbo Mode (TURBO = OFF), and the device is inactive for approx. 75 ns, increase time by amount shown.
11. Measured with eight outputs switching. See t_{PD} vs. Number of Outputs Switching graph.
12. t_{PZX} and t_{PXZ} are measured at ±0.5V from steady state voltage as driven by spec. output load. t_{PXZ} is measured with C_L = 5 pF.

REGISTER MODE—SYNCHRONOUS CLOCK A.C. CHARACTERISTICS

T_A = 0°C to 70°C, V_{CC} = 5.0V ±5%(9)

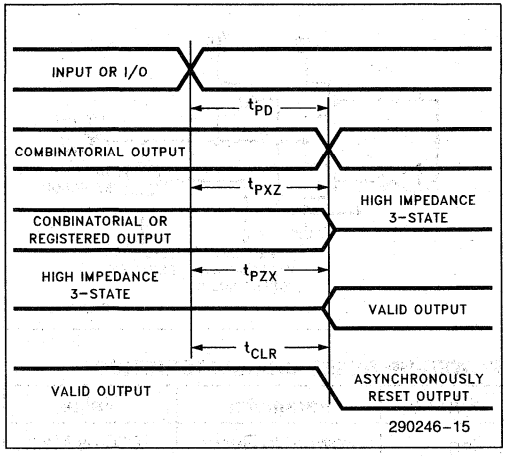
Symbol	Parameter	85C060-12			85C060-15			85C060-25			Non-Turbo(10) Mode	Units
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
f_{CNT1}(11)	Max. Counter Frequency $1/(t_{SU} + t_{CO1})$—Ext. Feedback	66	75		50	66		40	50			MHz
f_{CNT2}(11)	Max. Counter Frequency $1/(t_{CNT})$—Internal Feedback	83.3	90		66	75		40	66			MHz
f_{MAX}(11)	Max. Frequecy (Pipelined) $1/(t_{CW})$—No Feedback	100	111		83.3	100		66	80			MHz
t_{SU}	Input or I/O Setup Time to CLK	8			12			15			+25	ns
t_H	Input or I/O Hold Time from CLK	0			0			0				ns
t_{CO1}(11)	CLK High to Output Valid			7			8			10		ns
t_{CO2}(11)	CLK High to Output Valid Fed through Comb. Macrocell			16			20			30	+25	ns
t_{CNT}(11)	Macrocell Output Feedback to Macrocell Input—Internal Path			12			15			25	+25	ns
t_{CL}	CLK Low Time	4			5			6				ns
t_{CH}	CLK High Time	4			5			6				ns
t_{CW}	CLK Width	10			12			15				ns

REGISTER MODE—ASYNCHRONOUS CLOCK A.C. CHARACTERISTICS

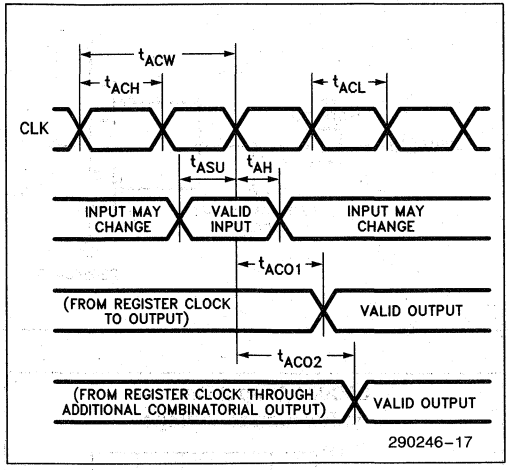
T_A = 0°C to 70°C, V_{CC} = 5.0V ±5%(9)

Symbol	Parameter	85C060-12			85C060-15			85C060-25			Non-Turbo(10) Mode	Units
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
f_{ACNT1}(11)	Max. Counter Frequency $1/(t_{ASU} + t_{ACO1})$—Ext. Feedback	58.8	70		50	66		33.3	40			MHz
f_{ACNT2}(11)	Max. Counter Frequency $1/(t_{ACNT})$—Internal Feedback	83.3	90		66	75		40	50			MHz
f_{AMAX}(11)	Max. Frequency (Pipelined) $1/(t_{ACW})$—No Feedback	83.3	90		66	75		50	66			MHz
t_{ASU}	Input or I/O Setup Time to Asynch. CLK	3			4			5			+25	ns
t_{AH}	Input or I/O Hold Time from Asynch. CLK	5			6			8				ns
t_{ACO1}(11)	Asynch. CLK High to Output Valid			14			16			25	+25	ns
t_{ACO2}(11)	Asynch. CLK High to Output Valid Fed through Comb. Macrocell			25			30			45	+25	ns
t_{ACNT}	Macrocell Output Feedback to Macrocell Input—Internal Path			12			15			25	+25	ns
t_{ACL}	Asynch. CLK Low Time	5			6			7				ns
t_{ACH}	Asynch. CLK High Time	5			6			7				ns
t_{ACW}	Asynch. CLK Width	12			15			20				ns

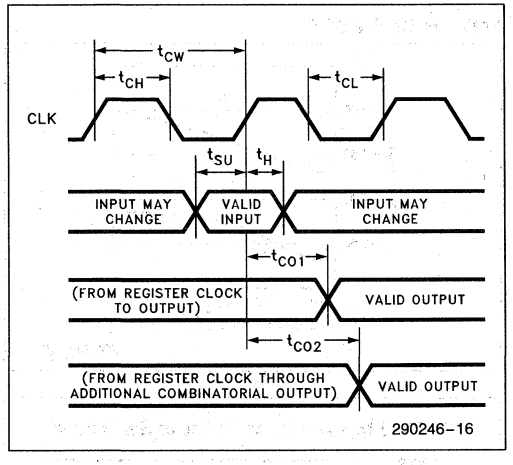
COMBINATORIAL MODE



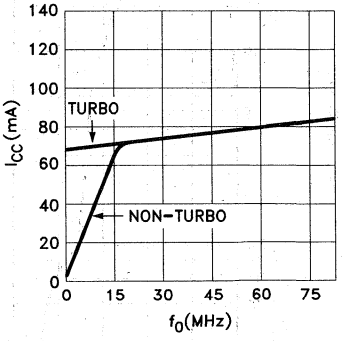
ASYNCHRONOUS REGISTERED MODE



SYNCHRONOUS REGISTERED MODE



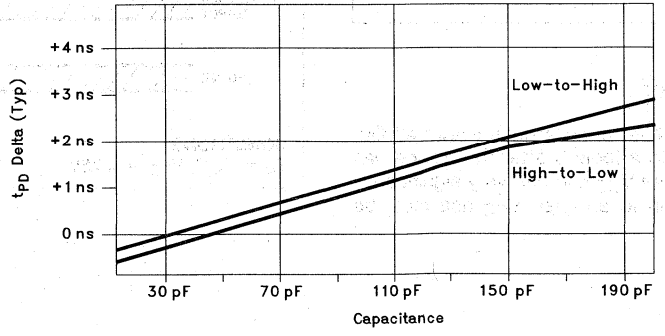
I_{CC} vs Frequency



290246-18

Conditions: T_A = 25°C, V_{CC} = 5.0V

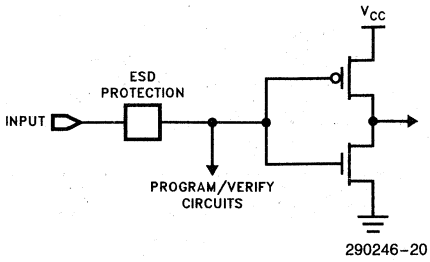
t_{PD} Derating vs Capacitive Loading



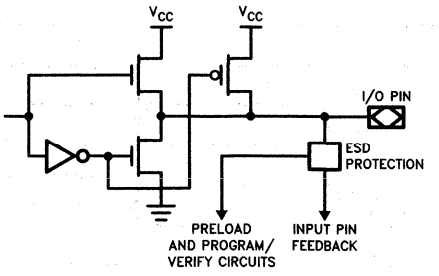
290246-19

Conditions: T_A = +70°C; V_{CC} = 4.75V

INPUT/OUTPUT EQUIVALENT SCHEMATICS

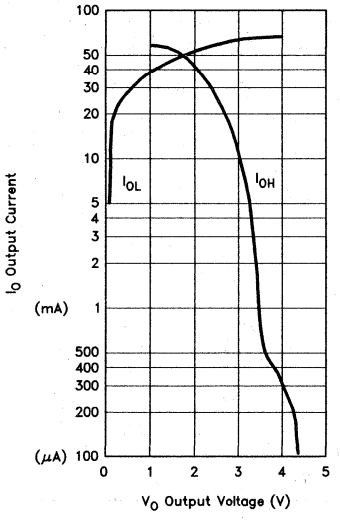


290246-20



290246-21

85C060 Output Current in Relation to Voltage



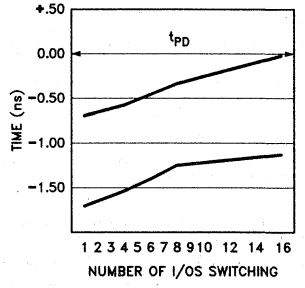
290246-24

CONDITIONS:
T_A = +80°C
V_{CC} = 4.75V

POWER-UP RESET CHARACTERISTICS

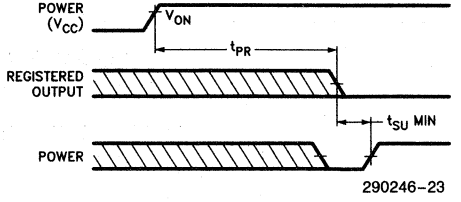
Symbol	Parameter	Value
t_{PR}	Power-Up Reset	1000 ns Max.
V_{ON}	Turn-On Voltage	4.75V

POWER-UP RESET



290246-22

85C060 t_{PD} vs No. of Outputs Switching



290246-23

CONDITIONS:
T_A = 70°C, V_{CC} = 4.75V, C_L = 30 pF

Power-Up Reset

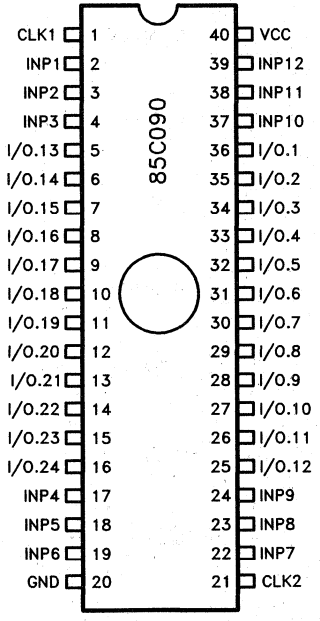
Internal power-up reset circuits ensure that all flip-flops will be reset to a logic 0 after the device has powered up. Because V_{CC} rise can vary significantly from one application to another, V_{CC} rise must be monotonic.

85C090
24-MACROCELL
CHMOS μPLD

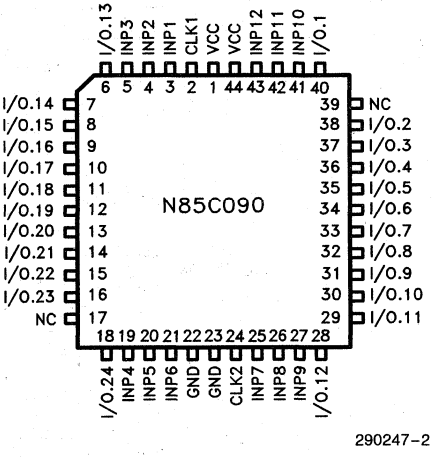
- High-Performance LSI Semi-Custom Logic Alternative to Low-End Gate Arrays, TTL, 74HC SSI and MSI Logic, and Bipolar PLDs
- High Speed Upgrade to 85C090, EP900, EP910 and EP930
- t_{PD} 15 ns, 50 MHz w/Feedback, Clock to Output 9 ns
- Typical I_{CC} = 105 mA @ 50 MHz
- 24 Macrocells with Programmable I/O Architecture (Register/Combinatorial). Registers Configurable as D/T/JK/RS Types
- Up to 26 Inputs (12 Dedicated and 24 I/O)
- 8 P-terms, Selectable SOP Invert, Clear and OE P-terms for Each Macrocell
- Programmable Clock System with 2 Synchronous Clocks and Asynchronous Clocking Option on all Macrocells
- 1-Micron CHMOS IIIE* EPROM Technology, UV-Erasable (CerDIP) or OTP
- Programmable Low-Power Option for "Standby" Operation; 60 μA Typ. in Standby Mode
- Programmable Security Bit Allows Total Protection of Proprietary Designs
- 100% Generically Tested Logic Array
- Available in 40-Pin CerDIP/PDIP and 44-Pin PLCC Packages

(See Packaging Spec., Order Number #231369)

2



290247-1



290247-2

Figure 1. Pinout Diagrams

*CHMOS is a patented process of Intel Corporation.

October 1990
Order Number: 290247-002

INTRODUCTION

The 85C090 is a high-performance, high-integration, general-purpose CMOS PLD. The 85C090 μPLD (Microcomputer Programmable Logic Device) accommodates logic functions with up to 36 inputs and 24 I/O macrocells. Each I/O macrocell includes 8 p-terms for input, a separate clear p-term, and an output enable/asynchronous clock p-term. With a maximum external frequency of 50 MHz, the 85C090 is well suited to high-performance microprocessor-based systems. The 85C090 is pin- and function-compatible with the 5C090, EP900, EP910 and EP930.

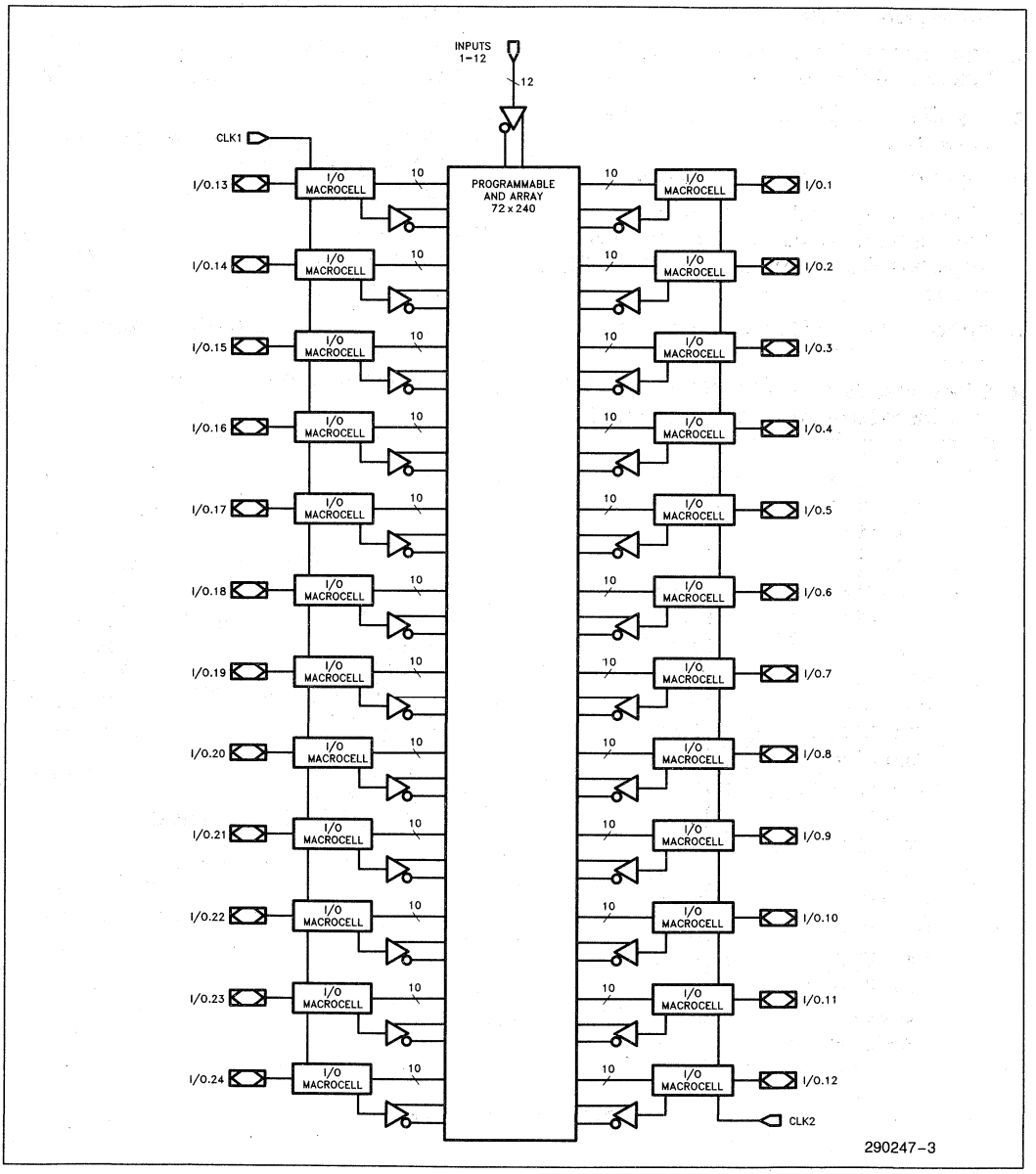


Figure 2. 85C090 Global Architecture

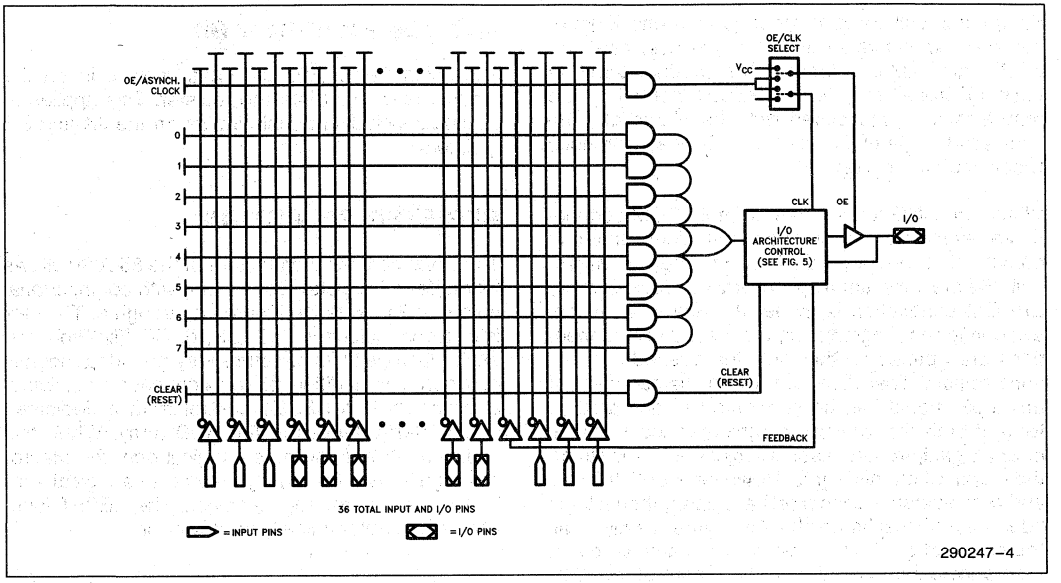


Figure 3. 85C090 Macrocell Architecture

The 85C090 uses CHMOS EPROM (floating gate) cells as logic control elements instead of fuses. The CHMOS EPROM technology reduces power consumption without sacrificing speed. In addition, Intel's advanced CHMOS III-E EPROM process technology enables higher logic densities to be achieved with superior speed and low-power performance over other comparable devices. Intel's μPLDs add the benefits of "zero" stand-by power not available on other programmable logic devices. EPROM technology allows these devices to be 100% factory tested by programming and erasing all the EPROM logic control elements.

The architecture of the 85C090 is based on the "Sum of Products" PLA (Programmable Logic Array) structure with a programmable AND array feeding into a fixed OR array. The device accommodates combinational and sequential logic functions. A programmable I/O architecture provides individual selection of either combinatorial or registered output and feedback signals all with selectable polarity.

A feature unique to the 85C090 is the ability to individually program the output registers as a D-, T-, SR-, or JK-type Flip-Flop without sacrificing the utilization of programmable AND logic. Each output register can be individually clocked from any of the input or feedback paths available within the AND array. With these features, a wide variety of logic functions can be simultaneously implemented—all on the same device.

ARCHITECTURE DESCRIPTION

The 85C090 has 12 dedicated inputs, 24 I/O pins that may be configured for input, output, or bidirectional operations, and 2 synchronous clock inputs. The 85C090 is packaged in a 40-lead windowed ceramic DIP or OTP plastic or 44-lead OTP J-leaded chip carrier package.

The basic Macrocell architecture for the 85C090 is shown in Figure 3. The 85C090 has 24 of these macrocells (one for each I/O pin). The Macrocell is organized in the familiar sum-of-products structure with a programmable AND array attached to a fixed OR term. The inputs to the programmable AND array originate from the true and complement signals from each of the dedicated input pins and each of the I/O control blocks.

The AND array for the 85C090 has 72 inputs derived from the true and complement signals at the input and I/O pins. The AND array in the 85C090 encompasses 240 product terms which are distributed among the 24 Macrocells. The global device architecture is shown in Figure 2.

Each Macrocell contains ten product terms. Eight of the ten product terms (AND gates) are dedicated for SOP logic implementation. One product term on each Macrocell is used for RESET control to the output register associated with the Macrocell. The final product term is used for OE/Asynchronous Clock implementation.

Within the AND array, there is an EPROM connection at every intersection of an input signal (true and complement) and a product term to a given Macrocell. Before programming an erased device, every EPROM connection is made at every intersection. But during the programming process, these connections are opened so that only the desired connections remain. Therefore, the true or complement of any input signal can be connected to any product term. If both the true and complement connections of any signal are left intact, a logical false results on the output of the AND gate. However, if both the true and complement connections are open, then a logic "don't care" results on the AND gate. Lastly, if all the inputs of a product term are programmed open, then a logical true results on the output of the AND gate.

The 85C090 has two dedicated clock inputs to provide synchronous clock signals to the internal registers. Each of the clock signals controls half the total registers within the given device. For example, CLK1 provides synchronous clocking to the registers in Macrocells in the left half of the array while CLK2 controls the registers associated with Macrocells in the right half of the array. The advanced I/O architecture allows for any number of the registers to be synchronously clocked (from none to all). Both of the dedicated clock inputs latch the data into a given register when triggered on a positive edge.

MACROCELL ARCHITECTURE SELECTION

The 85C090 architecture provides each Macrocell with over 50 different possible I/O register configurations. Each I/O pin can be configured for combinatorial or registered output (true or complement) with feedback. In addition, four different types of output registers can be implemented on I/O pin without any additional logic requirements. The feedback mechanism for each register back into the AND array can be programmed to provide for either registered feedback from the Macrocell or input feedback (treating the pin as an input). Another advantage of the advanced I/O capability of the 85C090 is the ability to individually clock each internal register from asynchronous clock signals.

Invert Select EPROM Bit

The Invert Select EPROM bit is used to invert the product term input into the register. This applies to all inputs including double inputs on the JK and SR registers.

REGISTER SELECTION

The advanced I/O architecture of the 85C090 allows four different register types along with combinatorial output as illustrated in Figure 4a through e. The register types include a T, D, JK, or SR Flip-Flop and each Macrocell I/O structure may be independently configured. In addition, all registers have an individual asynchronous RESET control from a dedicated product term derived in the AND array. When this dedicated product term is a logical one, the Macrocell register is immediately cleared to a logical zero independent of the register clock. The RESET function occurs automatically on power-up.

Output Register Configuration

The four different register types shown in Figure 4 are described below.

D- or T-type Flip-Flops

When either a D- or T-type Flip-Flop is configured as part of the I/O structure, all eight of the product terms into the Macrocell are ORed together and fed into the register input.

JK or SR Registers

When either a JK or SR register is configured, the eight product terms are shared among two OR gates (one for the J or S input and the other for the K or R input). The allocation for these product terms for each of the register inputs is optimized by the iPLS II software.

OUTPUT/FEEDBACK

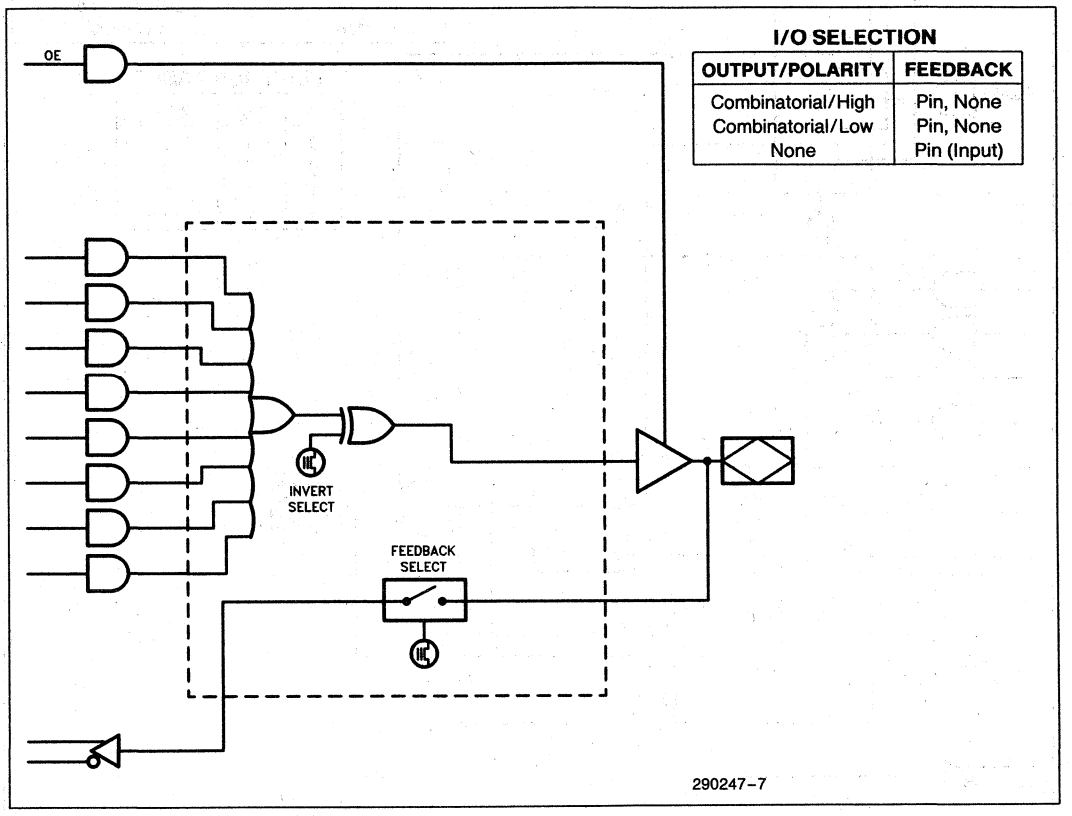
The Output Select Multiplexer allows for either registered, combinatorial or no output.

The Feedback Select Multiplexer EPROM bit enables registered, I/O (using the pin for bidirectional input or just input), or no feedback to the AND array.

The Feedback Select is also important for building product terms with more than 8 products. The 8-product product term of a Macrocell can be fed back

into the AND array and combined with still more signals to create a much larger product term (of more than 8-inputs). In addition, if the feedback product term is not to be output, then the iPLS II will reserve the associated Macrocell pin and indicate it in the REPORT file. A reserved pin should be left floating (no connect) when assembled onto a circuit board.

Any I/O pin may be configured as a dedicated input by selecting no output and pin feedback through the appropriate multiplexers.



I/O SELECTION

OUTPUT/POLARITY	FEEDBACK
Combinatorial/High	Pin, None
Combinatorial/Low	Pin, None
None	Pin (Input)

Figure 4a. Combinatorial I/O Configuration

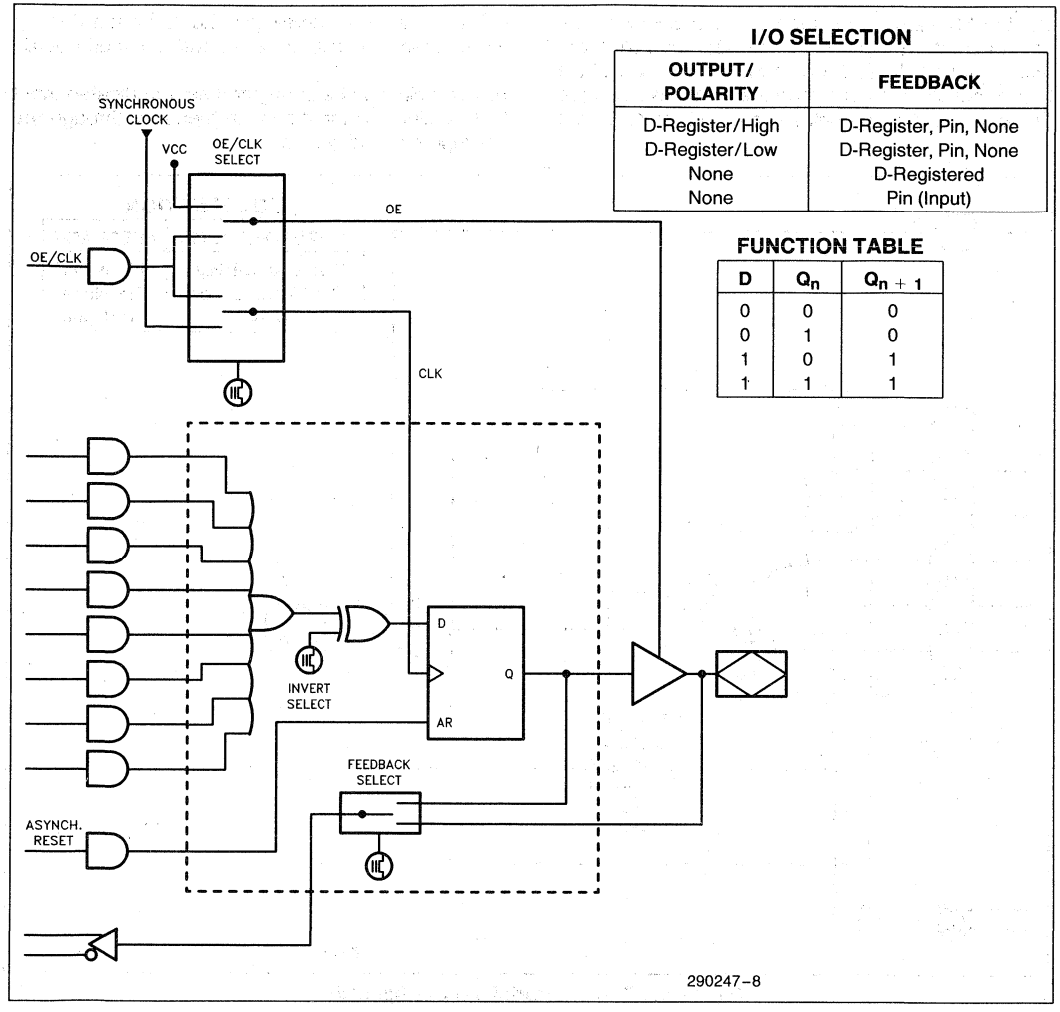


Figure 4b. D-Type Flip-Flop Register Configuration

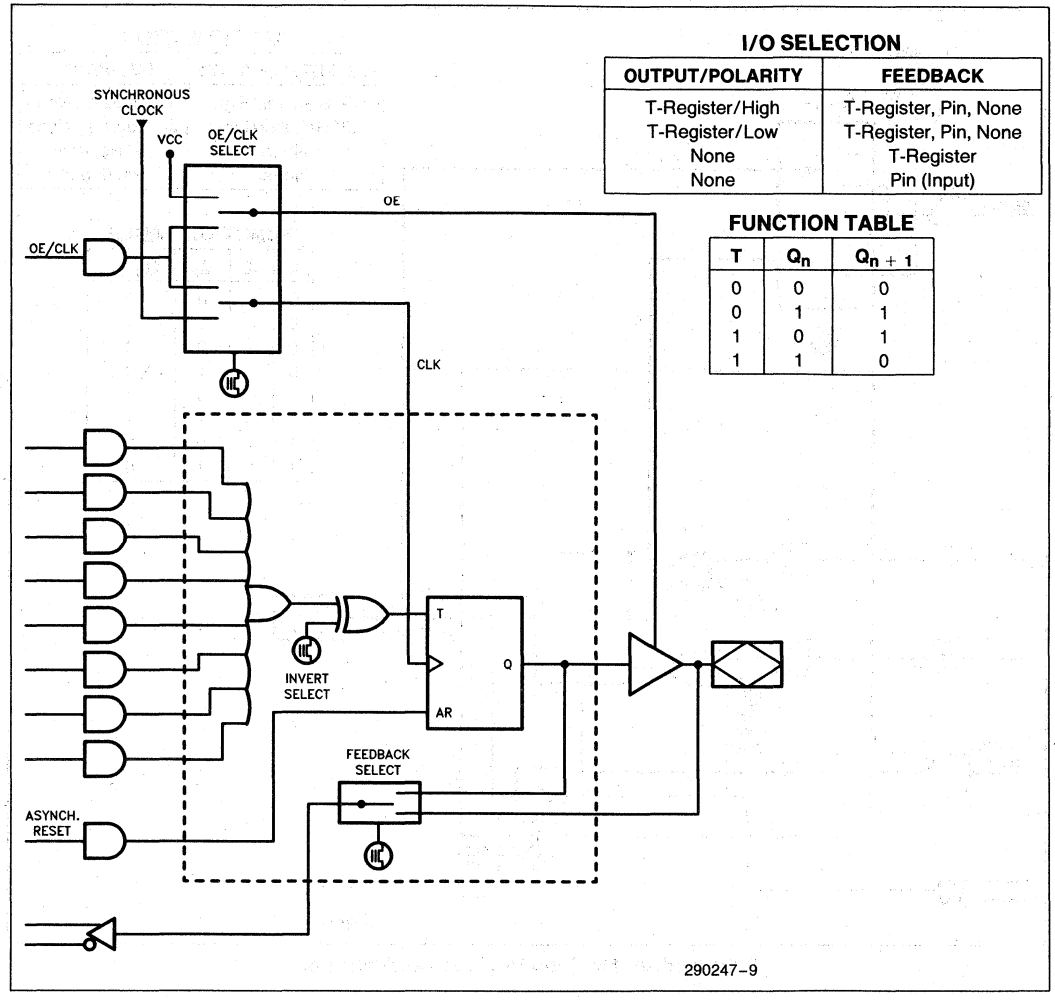
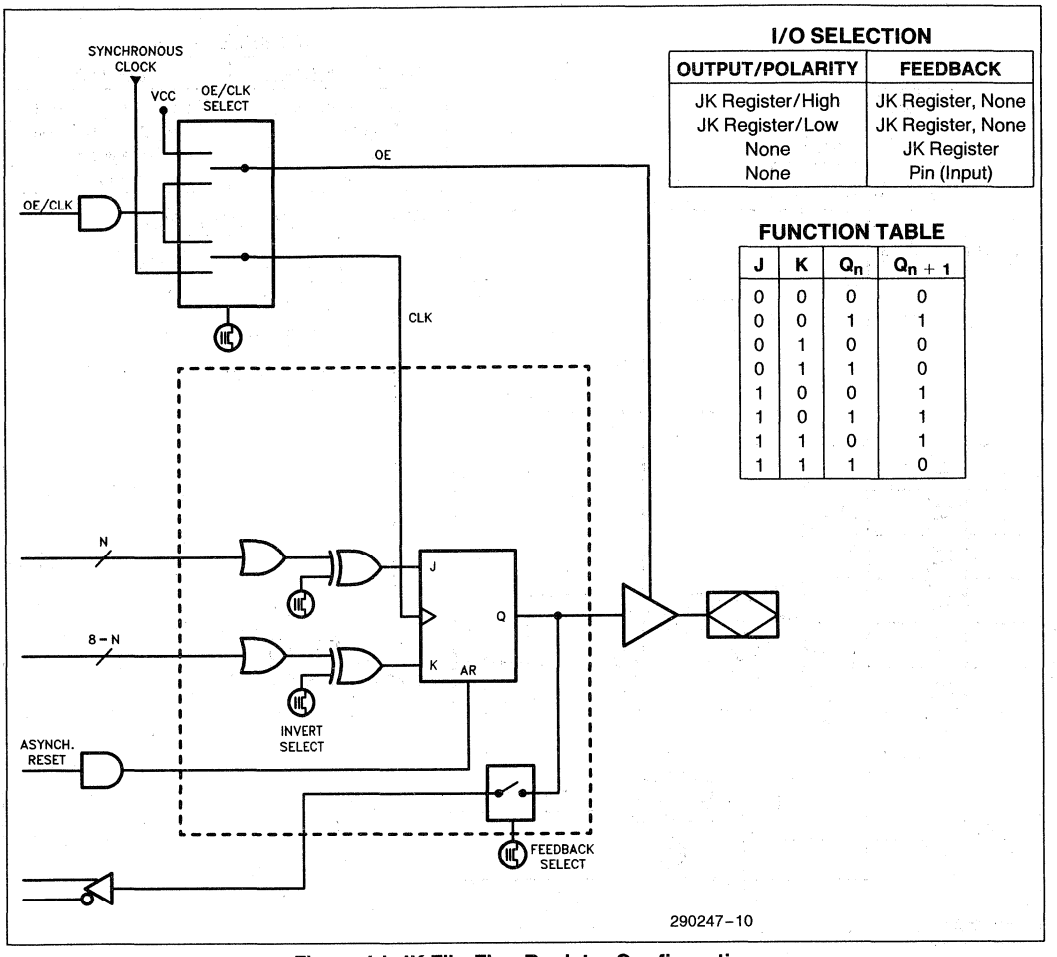


Figure 4c. Toggle Flip-Flop Register Configuration



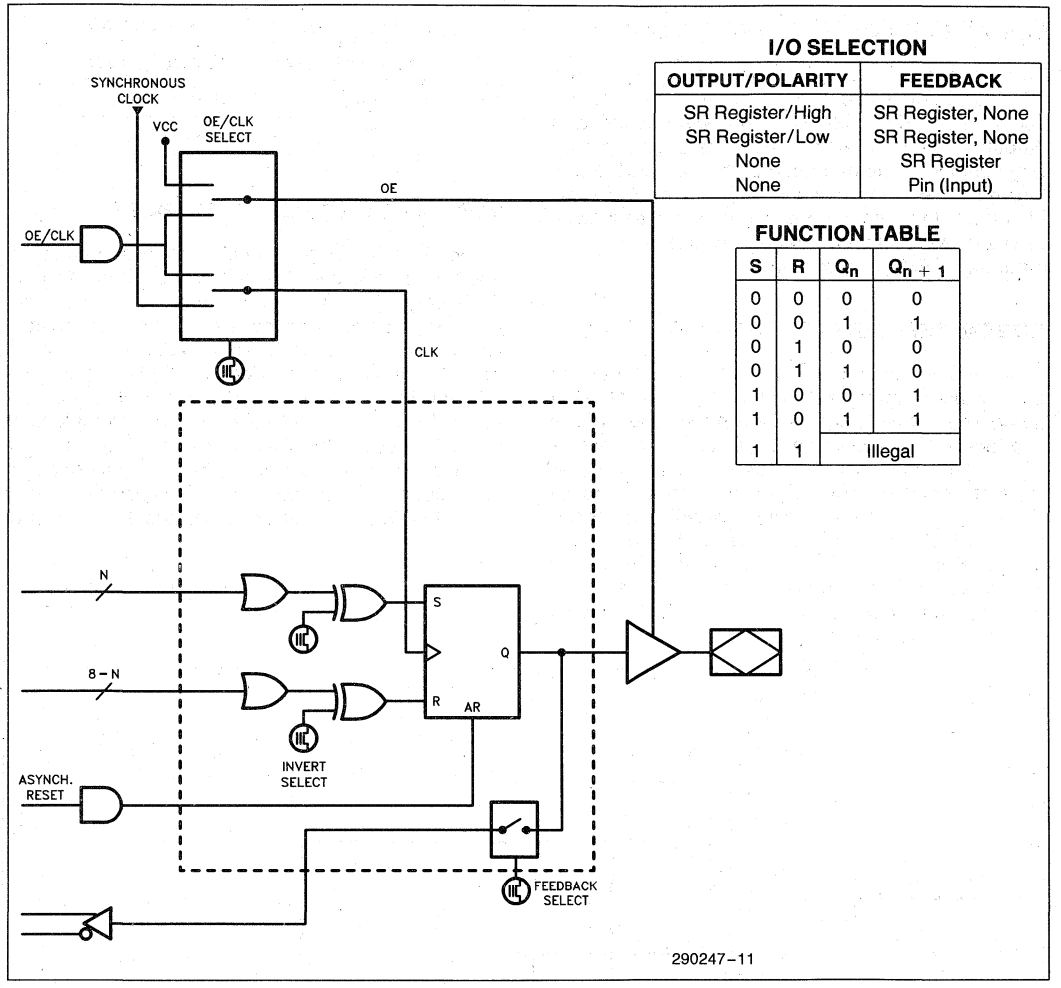
I/O SELECTION

OUTPUT/POLARITY	FEEDBACK
JK Register/High	JK Register, None
JK Register/Low	JK Register, None
None	JK Register
None	Pin (Input)

FUNCTION TABLE

J	K	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Figure 4d. JK Flip-Flop Register Configuration



I/O SELECTION

OUTPUT/POLARITY	FEEDBACK
SR Register/High	SR Register, None
SR Register/Low	SR Register, None
None	SR Register
None	Pin (Input)

FUNCTION TABLE

S	R	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	Illegal	

Figure 4e. SR Flip-Flop Register Configuration

Output Enable (OE)/Clock Selection

Two modes of operation are provided by the OE/CLK Select Multiplexer as a part of each Macrocell. One mode provides for three-state buffering of outputs while in the other mode, the outputs are always enabled. The operation of the OE/CLK Select Multiplexer sets the mode within a given Macrocell. Therefore, the output mode can be selected individually on every output. Figure 5 illustrates the two modes of OE/CLK operation.

MODE 0: THREE-STATE BUFFERING

In Mode 0, the three-state output buffer is controlled by a single product term originating from the AND array. The output is enabled when the product term is a logical true. Conversely, the output appears as high impedance when the product term is a logical false as shown in Table 1. In Mode 0, the Macrocell Flip-Flop is connected to its associated synchronous clock (either CLK1 or CLK2 depending upon the Macrocell's location within the device). Thus, the Macrocell Flip-Flop may be clocked by its respective synchronous clock but its output will not become valid until the output is enabled.

Table 1. Mode 0 Output Selection

Product Term	Output Buffer
FALSE	Three-State
TRUE	Enabled

MODE 1: OUTPUT BUFFER ENABLED

In Mode 1, the Output Buffer is always enabled. In addition, the Macrocell Flip-Flop is connected to the AND array. The Macrocell Flip-Flop may now be triggered from an asynchronous clock signal generated by the AND array logic to the OE/CLK multiplexable term. Mode 1 allows the Macrocell Flip-Flops to be individually clocked from any of the available signals in the AND array. Since both true and complement values appear in the AND array, the Flip-Flop may be clocked by positive-or negative-going signals at any input pin. Gated clock structures can be created since the Flip-Flop clock is created by a product term.

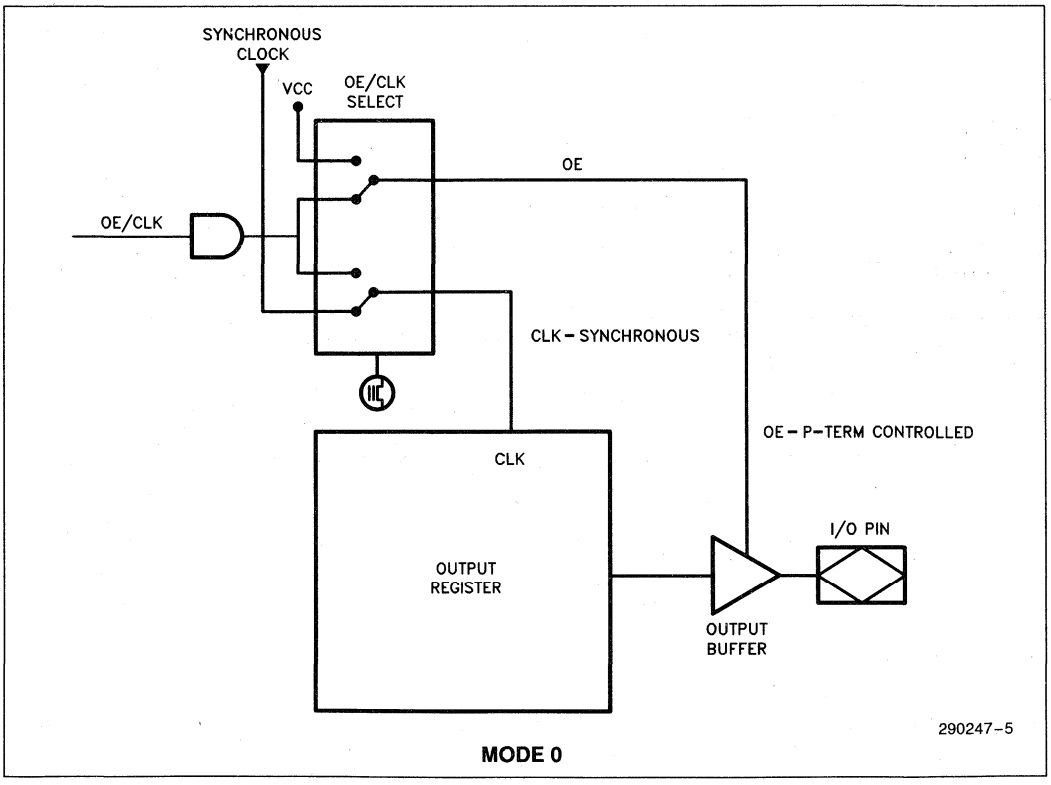


Figure 5. Output Enable/Clock Configuration

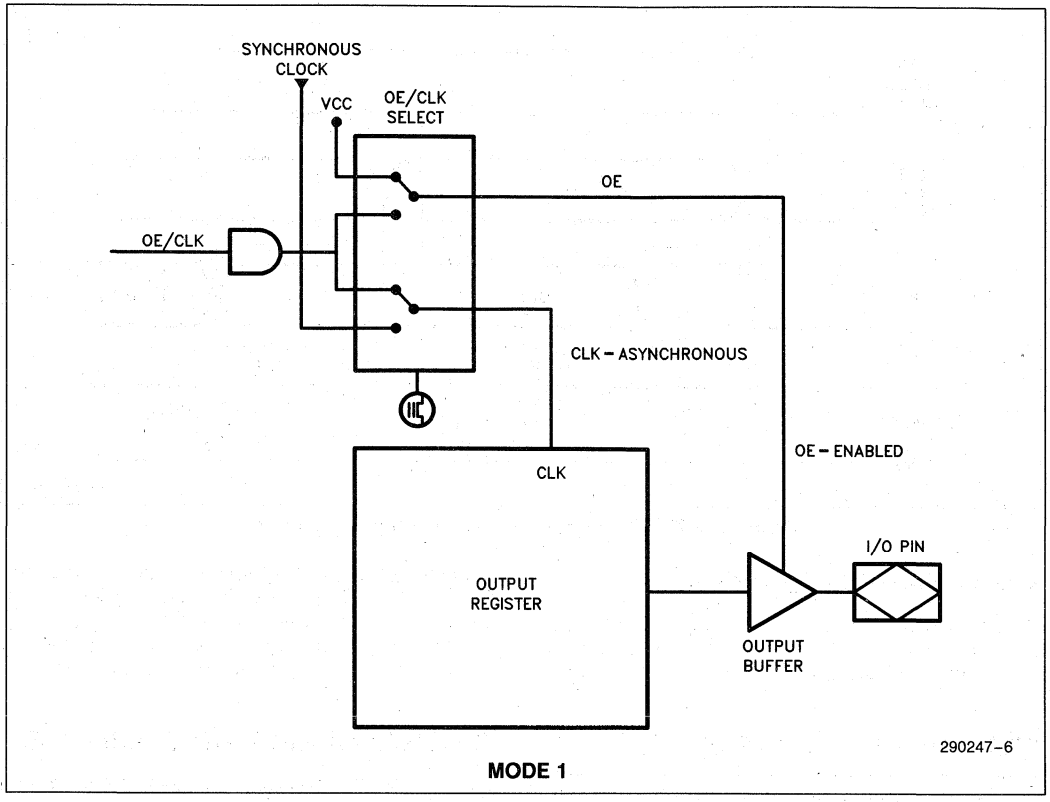


Figure 5. Output Enable/Clock Configuration (Continued)

AUTOMATIC STAND-BY MODE

The 85C090 contains a programmable bit, the Turbo Bit, that optimizes operation for speed or for power savings. When the Turbo Bit is programmed (TURBO = ON), the device is optimized for maximum speed. When the Turbo Bit is not programmed (TURBO = OFF), the device is optimized for power savings by entering standby mode during periods of inactivity.

Figure 6 shows the device entering standby mode approximately 75 ns after the last input transition. When the next input transition is detected, the device returns to active mode. Wakeup time adds an additional 40 ns to the propagation delay through the device as measured from the first input. No delay will occur if an output is dependent on more than one input and the last of the inputs changes after the device has returned to active mode.

After erasure, the Turbo Bit is unprogrammed (OFF); automatic standby mode is enabled. When the Turbo Bit is programmed (ON), the device never enters standby mode.

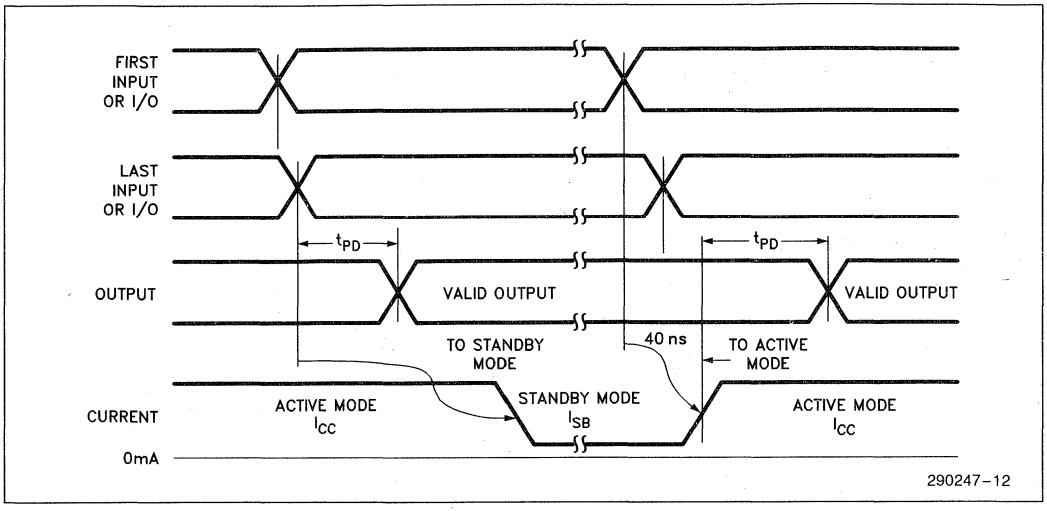


Figure 6. 85C090 Standby and Active Mode Transitions

Erased-State Configuration

Prior to programming or after erasing, the I/O structure is configured for combinatorial active low output with input (pin) feedback.

ERASURE CHARACTERISTICS

Erase time for the 85C090 is 1 hour at 12,000 $\mu W/cm^2$ with a 2537Å lamp.

Erasure characteristics of the device are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000Å. It should be noted that sunlight and certain types of flourescent lamps have wavelengths in the 3000–4000Å. Data shows that constant exposure to room level flourescent lighting could erase the typical device in approximately six years, while it would take approximately two weeks to cause erasure when exposed to direct sunlight. If the 85C090 is to be exposed to these types of lighting conditions for extended periods of time, conductive opaque labels should be placed over the device window to prevent unintentional erasure.

The recommended erasure procedure for the 85C090 is exposure to shortwave ultraviolet light with a wavelength of 2537Å. The integrated dose (i.e., UV intensity × exposure time) for erasure should be a minimum of forty (40) $Wsec/cm^2$. The erasure time with this dosage is 1 hour using an ultraviolet lamp with a 12,000 $\mu W/cm^2$ power rating. The 85C090 should be placed within one inch of the lamp tubes during erasure. The maximum integrated dose the 85C090 can be exposed to without damage is 7258 $Wsec/cm^2$ (1 week at 12,000 $\mu W/cm^2$). Exposure to high intensity UV light for longer periods may cause permanent damage to the device.

PROGRAMMING CHARACTERISTICS

Initially, and after erasure, all the EPROM control bits of the 85C090 are connected (in the "1" state). Each of the connected control bits are selectively disconnected by programming the EPROM cells into their "0" state. Programming voltage and waveform specifications are available by request from Intel to support programming of the 85C090.

int$_e$ligent Programming™ Algorithm

The 85C090 supports the int$_e$ligent Programming Algorithm which rapidly programs Intel PLDs while ensuring programming reliability as the incremental program margin of each bit is continually monitored to determine when the bit has been successfully programmed.

FUNCTIONAL TESTING

Since the logical operation of the 85C090 is controlled by EPROM elements, the device is completely testable. Each programmable EPROM bit controlling the internal logic is tested using application-independent test program patterns. After testing, the devices are erased before shipment to customers. No post-programming tests of the EPROM array are required.

The testability and reliability of EPROM-based programmable logic devices is an important feature over similar devices based on fuse technology. Fuse-based programmable logic devices require a user to perform post-programming tests to insure proper programming. These tests must be done at the device level because of the cummulative error effect. For example, a board containing ten devices each possessing a 2% device fallout translates into an 18% fallout at the board level (it should be noted that programming fallout of fuse-based programmable logic devices is typically 2% or higher).

DESIGN RECOMMENDATIONS

For proper operation, it is recommended that all input and output pins be constrained to the voltage range GND $<$ (V_{IN} or V_{OUT}) $<$ V_{CC}. Unused inputs and I/Os should be tied to V_{CC} or GND to minimize device power consumption. Reserved pins (as indicated in the logic compiler REPORT file) should be left floating (no connect) so that the pin can attain the appropriate logic level. A power supply decoupling capacitor of at least 0.2 μF must be connected directly between V_{CC} and GND pins of the device.

As with all CMOS devices, ESD handling procedures should be used with the 85C090 to prevent damage to the device during programming, assembly and test.

DESIGN SECURITY

A single EPROM bit provides a programmable design security feature that controls the access to the data programmed into the device. If this bit is set, a proprietary design within the device cannot be copied. This EPROM security bit enables a higher degree of design security than fused-based devices since programmed data within EPROM cells is invisible even to microscopic evaluation. The EPROM security bit, along with all the other EPROM control bits, will be reset by erasing the device.

LATCH-UP IMMUNITY

All of the input, I/O, and clock pins of the 85C090 have been designed to resist latch-up which is inherent in inferior CMOS structures. The 85C090 is designed with Intel's proprietary CHMOS II-E EPROM process. Thus, each of the pins will not experience latch-up with currents up to $\pm$100 mA and voltages ranging from $-$1V to (V_{CC} + 1V). Furthermore, the programming pin is designed to resist latch-up to the 13.5V maximum device limit.

INTEL PROGRAMMABLE LOGIC DEVELOPMENT SYSTEM II (iPLDS II)

iPLDS II provides all the tools needed to design with Intel PLDs or compatible devices. In addition to providing development assistance, iPLDS II insulates the user from having to know all the intricate details of PLD architecture (the software will optimize a design to benefit from architectural features). It contains comprehensive third generation software that supports four different design entry methods, minimizes logic, does automatic pin assignments and produces the best design fit for the selected PLD. It is user friendly with guided menus, on-line Help messages and soft key inputs.

In addition, the iPLDS II contains programmer hardware in the form of an iUP-PC Universal Programmer Personal Computer to enable the user to program PLDs, read and verify programmed devices and also to graphically edit programming files. The software generates industry standard JEDEC object code output files which can be downloaded to other programmers as well.

iPLS II software interfaces to several schematic capture packages to enable designs to be entered in schematic form. IPLDview-286/IPLDdraw allows the designer to use familiar TTL symbols or EPLD design primitive symbols. User-defined symbols are also supported. IPLDdraw also provides a path to A.C. Timing simulation of EPLD designs.

SCHEMA III-PLD allows the designer to use TTL symbols, EPLD custom macros, or EPLD design primitive symbols. It also supports user-defined symbols.

Other design formats include Boolean equation entry (supported directly by iPLS II) and state machine entry (supported by iSTATE).

Detailed information on the Intel Programmable Logic Development System II is contained in a separate Intel data sheet. (Order Number: 280168). Refer to the tools section of the *Programmable Logic* handbook for complete information on development tools.

The 85C090 is also supported by third-party logic compilers such as ABEL*, CUPL*, PLDesigner*, Log/IC, etc. Programming support is provided by third-party programmer companies such as Data I/O, Logical Devices, STAG, etc. Please refer to the "Third-Party Support" lists in the *Programmable Logic* handbook for complete information and vendor contacts.

ADF PRIMITIVES SUPPORTED

The following ADF primitives are supported by this device:

INP	JOJF
CONF	JONF
COIF	SONF
RONF	SOSF
RORF	TOIF
ROIF	TONF
NORF	TOTF
NOJF	CLKB
NOSF	
NOTF	

ORDERING INFORMATION

f_{CNT1} (MHz)	f_{MAX} (MHz)	t_{PD} (ns)	Order Code	Package	Operating Range
50	83.3	15	D85C090-15	*CerDIP	Commercial
			P85C090-15	PDIP	
			N85C090-15	PLCC	
40	66	20	D85C090-20	*CerDIP	Commercial
			P85C090-20	PDIP	
			N85C090-20	PLCC	
33.3	50	25	D85C090-25	*CerDIP	Commercial
			P85C090-25	PDIP	
			N85C090-25	PLCC	

*Windowed package allows UV erase.

*ABEL is a trademark of Data I/O, Corporation. CUPL is a trademark of Logical Devices, Inc. PLDesigner is a trademark of MINC, Inc.

ABSOLUTE MAXIMUM RATINGS*

Symbol	Parameter	Min	Max	Units
V_{CC}	Supply Voltage(1)	−2.0	7.0	V
V_{PP}	Programming Supply Voltage(1)	−2.0	13.5	V
V_I	DC Input Voltage(1)(2)	−0.5	V_{CC} + 0.5	V
t_{stg}	Storage Temperature	−65	+150	°C
t_{amb}	Ambient Temperature(3)	−10	+85	°C

NOTES:
1. Voltages with respect to ground.
2. Minimum DC input is −0.5V. During transitions, the inputs may undershoot to −2.0V or overshoot to +7.0V for periods less than 20 ns under no load conditions.
3. Under bias. Extended temperature versions are also available.

NOTICE: This data sheet contains preliminary information on new products in production. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Min	Max	Unit
V_{CC}	Supply Voltage	4.75	5.25	V
V_{IN}	Input Voltage	0	V_{CC}	V
V_O	Output Voltage	0	V_{CC}	V
T_A	Operating Temperature	0	+70	°C
t_R	Input Rise Time		500	ns
t_F	Input Fall Time		500	ns

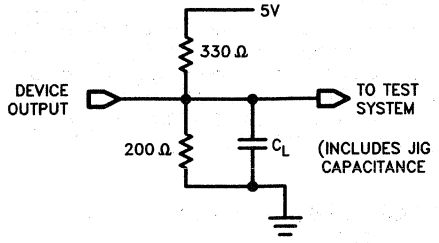
2

D.C. CHARACTERISTICS T_A = 0°C to +70°C, V_{CC} = 5.0V ±5%

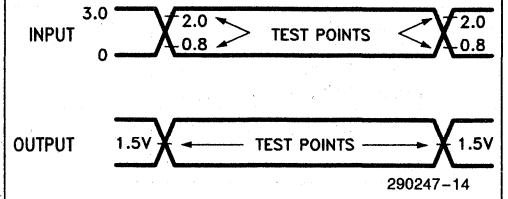
Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
V_{IH}(4)	High Level Input Voltage	2.0		V_{CC} + 0.3	V	
V_{IL}(4)	Low Level Input Voltage	−0.3		0.8	V	
V_{OH}	High Level Output Voltage	2.4			V	I_O = −4.0 mA D.C., V_{CC} = min.
V_{OL}(5)	Low Level Output Voltage			0.45	V	I_O = 12.0 mA D.C., V_{CC} = min.
I_I	Input Leakage Current	−10		+10	µA	V_{CC} = max., GND < V_{IN} < V_{CC}
I_{OZ}	Output Leakage Current	−10		+10	µA	V_{CC} = max., GND < V_{OUT} < V_{CC}
I_{SC}(6)	Output Short Circuit Current	−30		−120	mA	V_{CC} = max., V_{OUT} = 0.5V
I_{SB}(7)	Standby Current		60	150	µA	V_{CC} = max., V_{IN} = V_{CC} or GND, Standby Mode
I_{CC}	Power Supply Current (See I_{CC} vs. Freq. Graph)		4	12	mA	V_{CC} = max., V_{IN} = V_{CC} or GND, No Load, f_{IN} = 1 MHz, Device Prog. as Two 12-Bit Counters, Turbo = Off
			105	125	mA	Turbo = On, f_{IN} = 50 MHz

NOTES:
4. Absolute values with respect to device GND; all over and undershoots due to system or tester noise are included.
5. Maximum DC I_{OL} for the device is 96 mA for CLK1 group I/O.1-I/O.12 and 96 mA for CLK2 group I/O.13-I/O.24.
6. Not more than 1 output should be tested at a time. Duration of that test should not exceed 1 second.
7. In Non-Turbo Mode (TURBO = OFF), device enters standby mode approximately 75 ns after the last input transition.

A.C. TESTING LOAD CIRCUIT



A.C. TESTING INPUT, OUTPUT WAVEFORM



A.C. Testing: Inputs are driven at 3.0V for a Logic "1" and 0V for a Logic "0". Timing Measurements are made at 2.0V for a Logic "1" and 0.8V for a Logic "0" on inputs. Outputs are measured at a 1.5V point. Device input rise and fall times are less than 3 ns.

CAPACITANCE T_A = 0°C to +70°C; V_{CC} = 5.0V ±5%(8)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
C_{IN}	Input Capacitance	V_{IN} = 0V, f = 1.0 MHz		5	8	pF
C_{IO}	I/O Capacitance	V_{OUT} = 0V, f = 1.0 MHz		6	8	pF
C_{CLK}	CLK Capacitance	V_{OUT} = 0V, f = 1.0 MHz		8	10	pF
C_{VPP}	V_{PP} Pin Capacitance	V_{PP} on CLK2, f = 1.0 MHz		10	12	pF

NOTES:
8. These values are evaluated at initial characterization and whenever design modifications occur that may affect capacitance.

COMBINATORIAL MODE A.C. CHARACTERISTICS

T_A = 0°C to +70°C, V_{CC} = 5.0V ±5%(9)

Symbol	Parameter	85C090-15			85C090-20			85C090-25			Non-(10) Turbo Mode	Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
t_{PD}(11)	Input or I/O to Output Valid w/ 8 Outputs Switching			15			20			25	+40	ns
t_{PZX}(12)	Input or I/O to Output Enable			18			23			28	+40	ns
t_{PXZ}(12)	Input or I/O to Output Disable			18			23			28	+40	ns
t_{CLR}	Input or I/O to Asynch. Reset			18			23			28	+40	ns

NOTES:
9. Typical values are at T_A = 25°C, V_{CC} = 5V, Active Mode.
10. If device is operated in Non-Turbo Mode (TURBO = OFF) and the device is inactive for approx. 75 ns, increase time by amount shown.
11. Measured with eight outputs switching. See t_{PD} vs. Number of Outputs Switching graph.
12. t_{PZX} and t_{PXZ} are measured at ±0.5V from steady state voltage as driven by spec. output load. t_{PXZ} is measured with C_L = 5 pF.

REGISTER MODE—SYNCHRONOUS CLOCK A.C. CHARACTERISTICS

T_A = 0°C to 70°C, V_{CC} = 5.0V ±5%(9)

Symbol	Parameter	85C090-15			85C090-20			85C090-25			Non-(10) Turbo Mode	Units
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
f_{CNT1}(11)	Max. Counter Frequency $1/(t_{SU} + t_{CO})$—Ext. Feedback	50	66		40	50		33	40			MHz
f_{CNT2}(11)	Max. Counter Frequency $1/(t_{CNT})$—Internal Feedback	66.6	75		50	66		40	50			MHz
f_{MAX}(11)	Max. Frequency (Pipelined) $1/(t_{CW})$—No Feedback	83.3	100		66	75		50	66			MHz
t_{SU}	Input or I/O Setup Time to CLK	11			13			16			+40	ns
t_H	Input or I/O Hold Time from CLK	0			0			0				ns
t_{CO1}(11)	CLK High to Output Valid			9			12			14		ns
t_{CO2}	CLK High to Output Valid Fed Through Comb. Macrocell			20			25			30	+40	ns
t_{CNT}(11)	Macrocell Output Feedback to Macrocell Input—Internal Path			15			20			25	+40	ns
t_{CL}	CLK Low Time	5			6			8				ns
t_{CH}	CLK High Time	5			6			8				ns
t_{CP}	CLK Period	12			15			20				ns

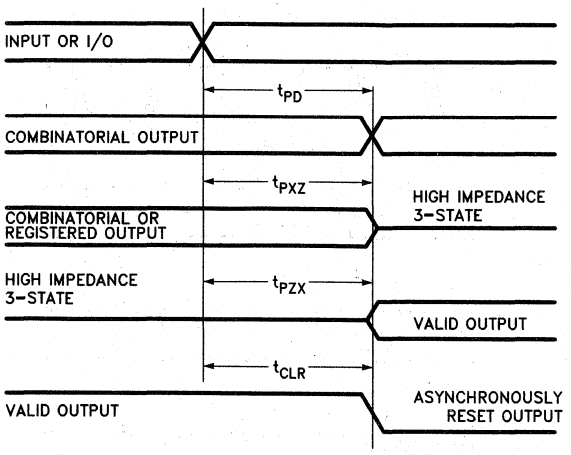
2

REGISTER MODE—ASYNCHRONOUS CLOCK A.C. CHARACTERISTICS

T_A = 0°C to 70°C, V_{CC} = 5.0V ±5%[8]

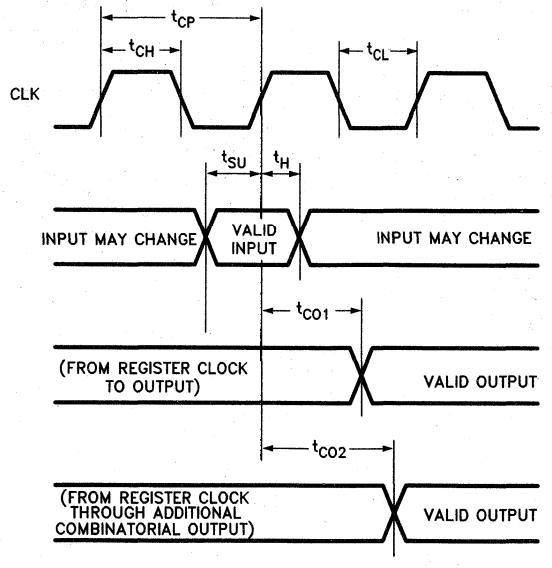
Symbol	Parameter	85C090-15			85C090-20			85C090-25			Non-[9] Turbo Mode	Units
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
f_{ACNT1}[10]	Max. Counter Frequency $1/(t_{ASU} + t_{ACO})$—Ext. Feedback	45.4	60		35.6	45		27.7	40			MHz
f_{ACNT2}[10]	Max. Counter Frequency $1/(t_{ACNT})$—Internal Feedback	66.6	80		50	66		40	50			MHz
f_{AMAX}[10]	Max. Frequency (Pipelined) $1/(t_{ACW})$—No Feedback	66.6	80		50	66		40	50			MHz
t_{ASU}	Input or I/O Setup to Asynch. CLK	4			5			8			+40	ns
t_{AH}	Input or I/O Hold from Asynch. CLK	6			7			8				ns
t_{ACO1}[10]	Asynch. CLK High to Output Valid			18			23			28	+40	ns
t_{ACO2}	Asynch. CLK High to Output Valid Fed Through Comb. Macrocell			29			36			44	+40	ns
t_{ACNT}[10]	Macrocell Output Feedback to Macrocell Input—Internal Path			15			20			25	+40	ns
t_{ACL}	Asynch. CLK Low Time	6			8			10				ns
t_{ACH}	Asynch. CLK High Time	6			8			10				ns
t_{ACP}	Asynch. CLK Period	15			20			25				ns

COMBINATORIAL MODE



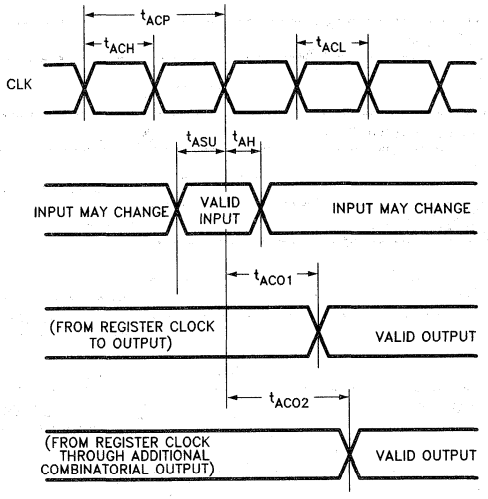
290247-15

SYNCHRONOUS REGISTERED MODE



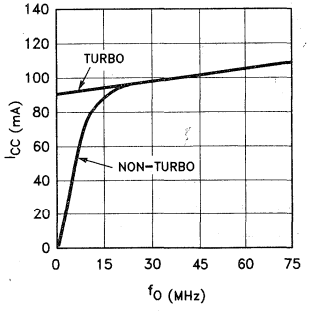
290247-16

ASYNCHRONOUS REGISTERED MODE



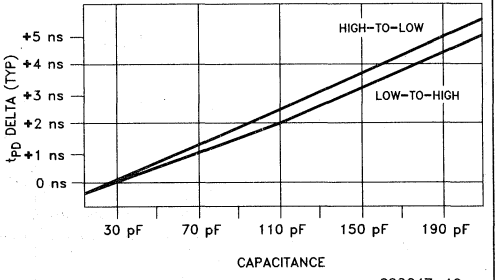
290247-17

85C090 I_{CC} vs Frequency



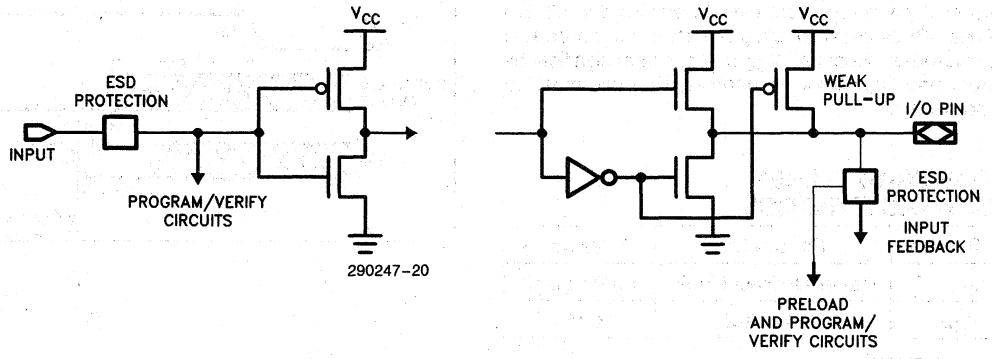
290247-18

85C090 t_{PD} Derating vs Capacitive Loading

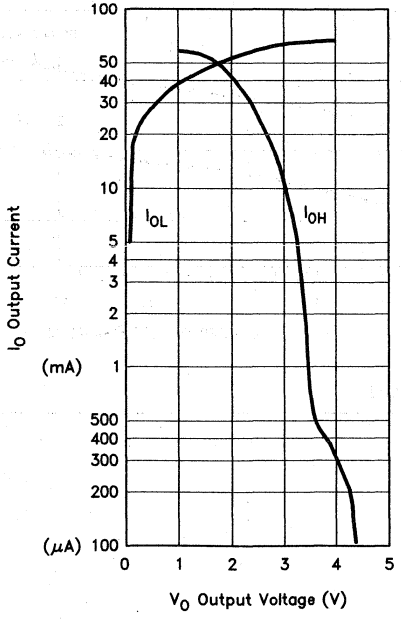


290247-19

Input/Output Equivalent Schematics



85C090 Output Current in Relation to Voltage



290247-24

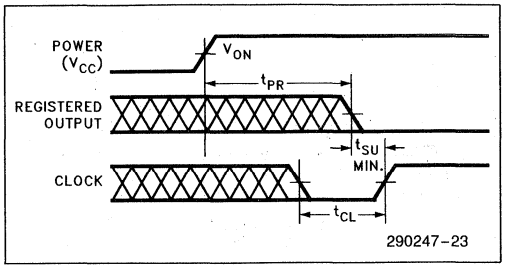
CONDITIONS:
T_A = +80°C
V_{CC} = 4.75V

2

Power-Up Reset

Internal power-up reset circuits ensure that all flip-flops will be reset to a logic 0 after the device has powered-up. Because V_{CC} rise can vary significantly from one application to another, V_{CC} rise must be monotonic.

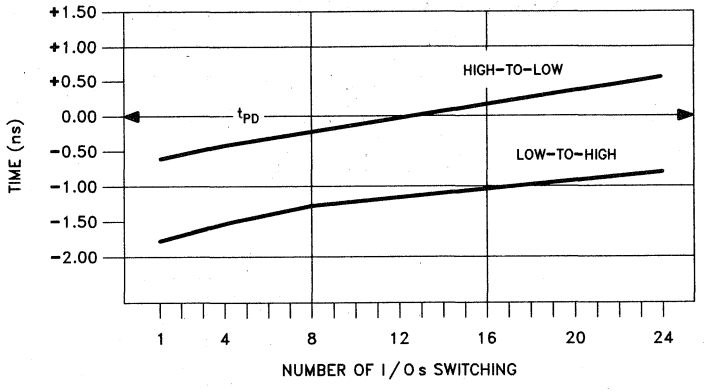
POWER-UP RESET



POWER-UP RESET CHARACTERISTICS

Symbol	Parameter	Value
t_{PR}	Power-Up Reset Time	1000 ns Max.
V_{ON}	Turn-On Voltage	4.75V

85C090 t_{PD} vs. No. of Outputs Switching



290247-22

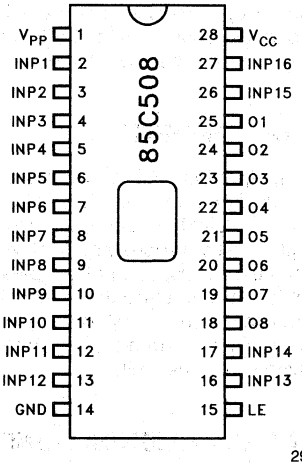
CONDITIONS:
T_A = 70°C, V_{CC} = 4.75V, Spec. Load

85C508
FAST 1-MICRON CHMOS DECODER/LATCH μPLD

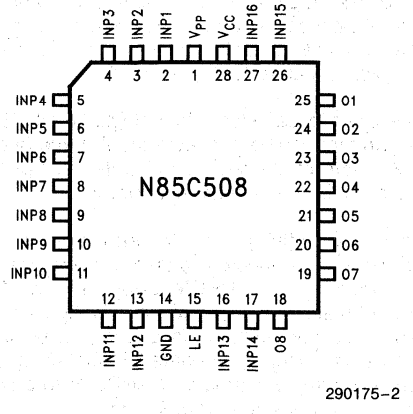
- **High-Performance Programmable Logic Device for High-Speed Microprocessor-to-Memory Decode**
- **Supports Intel386™, i468™, i860™, 80960 Series and Other High-Performance Systems**
- **Extremely High Speed—t_{PD} 7.5 ns (max), 133.3 MHz (max), t_{EO} 4.5 ns (max)**
- **Upgrade Alternative to Fast Bipolar PLDs and Fast MSI Logic**
- **16 Dedicated Inputs for Address/Data Bus Decoding; 8 Latched Outputs; 1 Global Latch Enable**
- **I_{CC} 15 mA Typ., 48 mA Max. @ 50 MHz**
- **100% Generically Testable Logic Array**
- **Available in 28-Pin 300-mil CerDIP/PDIP and PLCC Packages**

(See Packaging Spec., Order Number #231369)

2



290175-1



290175-2

Figure 1. Pinout Diagrams

October 1990
Order Number: 290175-004

INTRODUCTION

The 85C508 is a member of Intel's μPLD (Microcomputer Programmable Logic Device) family of devices. Produced on Intel's 1-micron CHMOS process, this device is designed to support the speeds required in fast microprocessor to memory paths. The sixteen inputs, p-term array, and eight output latches in the device provides address and data bus decoding and latching. The device takes full advantage of the lightning speed of Intel's 1-micron CHMOS technology. The device can be used as an upgrade to fast bipolar PLDs, and to fast AS, ALS, HC, or HCT SSI and MSI logic devices.

The 85C508 uses advanced EPROM cells as architecture and logic array memory elements instead of poly-silicon fuses. Coupled with Intel's proprietary CHMOS technology, the result is a device that offers a fast 7.5 ns t_{PD} in flow-through mode and a t_{EO} of 4.5 ns in latch mode. The inherent speed of the device makes it ideally suited for bus decoding applications with Intel386™, i486™, i860™, and 80960 systems. Output buffers on the device are designed to optimize signal transitions in high-speed applications (reduced overshoot and undershoot).

ARCHITECTURE DESCRIPTION

The architecture of the device is designed for high-speed performance, with dedicated inputs feeding a logic array. Outputs from the logic array feed the fast output latches. All output latches are controlled by the global LE (Latch Enable) signal. Figure 2 shows the global architecture of the 85C508.

The input to each latch is a single NAND (85C508) p-term that can be connected to the true or complement state of the dedicated inputs. All input signals are available to all eight macrocells.

Each intersecting point in the logic array is connected or not connected based on the value programmed in the EPROM array. Initially (EPROM erased state), no connections exist between any p-term and any input. Connections can be made by programming the appropriate EPROM cells. Since p-terms are implemented as NANDs. A true condition on a p-term drives the output low.

ERASURE CHARACTERISTICS

Erasure time for the 85C508 is 20 minutes at 12,000 μWsec/cm² with a 2537Å UV lamp.

Erasure characteristics of the device is such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000Å. It should be noted that sunlight and certain types of flourescent lamps have wavelengths in the 3000Å–4000Å range. Data shows that constant exposure to room level flourescent lighting could erase the typical device in approximately three years, while it would take approximately one week to erase the device when exposed to direct sunlight. If the device is to be exposed to these lighting conditions for extended periods of time, conductive opaque labels should be placed over the device window to prevent unintentional erasure.

The recommended erasure procedure for the 85C508 is exposure to shortwave ultraviolet light with a wavelength of 2537Å. The integrated dose (i.e., UV intensity x exposure time) for erasure should be a minimum of fifteen (15) Wsec/cm². The erasure time with this dosage is approximately 20 minutes using an ultraviolet lamp with a 12,000 μW/cm² power rating. The device should be placed within 1 inch of the lamp tubes during exposure. The maximum integrated dose the devices can be exposed to without damage is 7258 Wsec/cm² (1 week at 12,000 μW/cm²). Exposure to high intensity UV light for longer periods may cause permanent damage to the device.

POWER-UP

Internal power-up circuits ensure that the device will respond to inputs 1000 ns (max.) after V_{CC} reaches 4.75V. V_{CC} rise must be monotonic.

LATCH-UP IMMUNITY

All of the input, output, and clock pins of the device have been designed to resist latch-up, which is inherent in inferior CMOS structures. The device is designed with Intel's proprietary 1-micron CHMOS EPROM process. Thus, each of the pins will not experience latch-up with currents up to ±100 mA and voltages ranging from −0.5V to (V_{CC} + 0.5V). The programming pin is designed to resist latch-up to the 13.5V maximum device limit.

DESIGN RECOMMENDATIONS

For proper operation, it is recommended that all input and output pins be constrained to the voltage range GND ≤ (V_{IN} or V_{OUT}) ≤ V_{CC}. All unused inputs should be tied high or low to minimize power consumption (do not leave them floating). A power supply decoupling capacitor of at least 0.1 μF must be connected directly between each V_{CC} and GND pin. V_{PP} must be tied to GND.

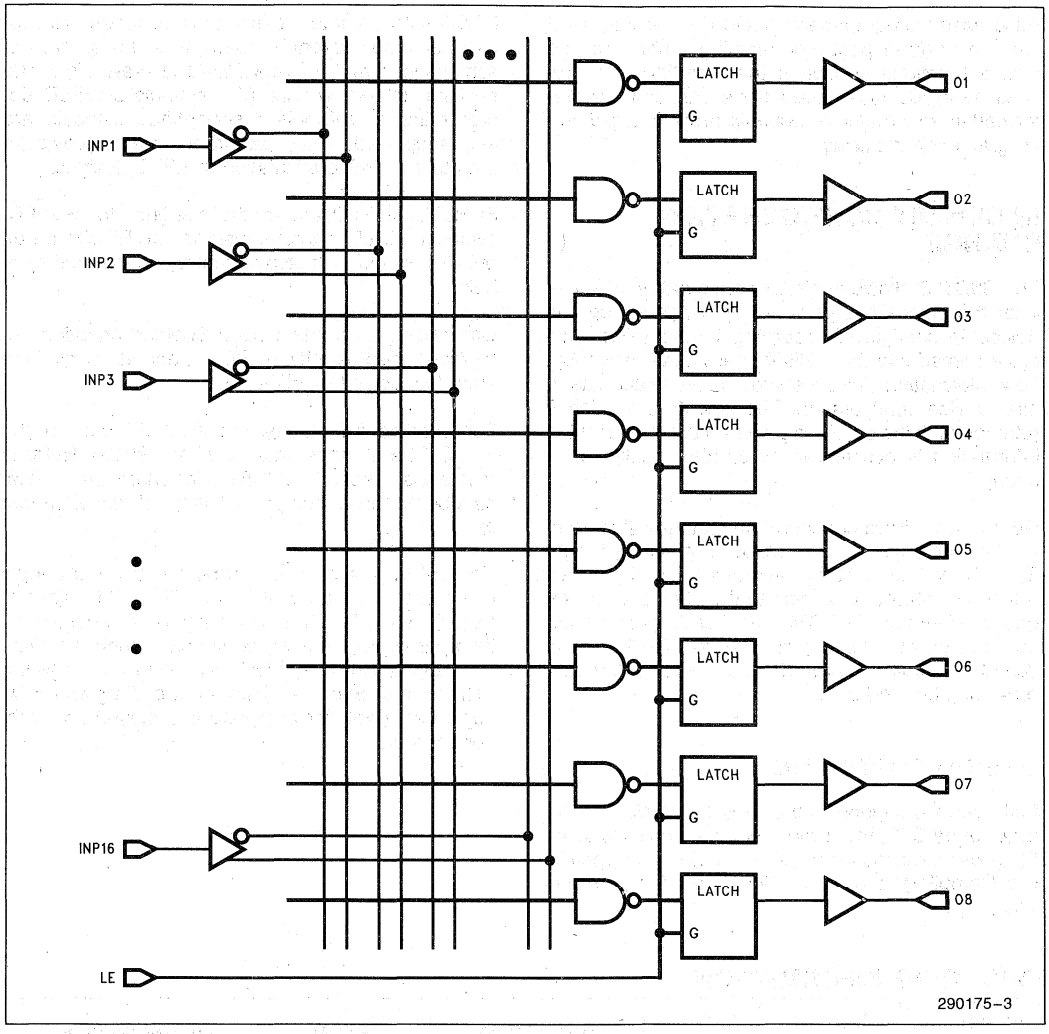


Figure 2. 85C508 Global Architecture

As with all CMOS devices, ESD handling procedures should be used with the 85C508 to prevent damage to the device during programming, assembly, and test.

FUNCTIONAL TESTING

Since the logical operation of the 85C508 is controlled by EPROM elements, the device is completely testable during the manufacturing process. Each programmable EPROM bit controlling the internal logic is tested using application independent test patterns. EPROM cells in the device are 100% tested for programming and erasure. After testing, the devices are erased before shipments to the customers. No post-programming tests of the EPROM array are required.

The testability and reliability of EPROM-based programmable logic devices is an important feature over similar devices based on fuse technology.

Fuse-based programmable logic devices require a user to perform post-programming tests to insure device functionality. During the manufacturing process, tests on fuse-based parts can only be performed in very restricted ways in order to avoid pre-programming the array.

IN-CIRCUIT CONFIGURATION CHANGE

The 85C508 allows in-circuit configuration changes after the device has powered up. At power-up, the device is configured according to the information programmed into the EPROM cells. After power-up, new information can be shifted in on select pins to alter device configuration. The new configuration is retained until the device is powered down or until the information is overwritten by another configuration change.

Note that in-circuit configuration changes allow "on-the-fly" changes to be made but do not alter EPROM cell data. At the next power-up, the device will be configured according to the original data programmed into the EPROM cells. For details on in-circuit configuration changes, refer to AP-337, *In-Circuit Reconfiguration of 85C960 and 85C508 MPLDs,* order number: 292072.

DESIGN SOFTWARE

Full software support is provided by version 2.0 (or later) of iPLS II (Intel Programmable Logic Software II). Those versions includes the LOC (Logic Optimizing Compiler) and APT (Advanced Programming Tool).

iPLS II software interfaces to several schematic capture packages to enable designs to be entered in schematic form. IPLDview-286/IPLDdraw allows the designer to use familiar TTL symbols or EPLD design primitive symbols. User-defined symbols are also supported. IPLDdraw products also provide a path to a.c. timing simulation of EPLD designs.

SCHEMA III-PLD allows the designer to use TTL symbols, EPLD custom macros, or EPLD design primitive symbols. It also supports user-defined symbols.

Other design formats include Boolean equation entry (supported directly by iPLS II) and state machine entry (supported by iSTATE).

For detailed information on iPLS II, refer to the iPLDS II Data Sheet, order number: 290134. Refer to the tools section of the *Programmable Logic* handbook for a complete listing of development tools.

The 85C508 is also supported by third-party logic compilers such as ABEL*, CUPL*, PLDesigner*, Log/IC, etc. Programming support is provided by third-party programmer companies such as Data I/O, Logical Devices, STAG, etc. Please refer to the "Third-Party Support" lists in the *Programmable Logic* handbook for complete information and vendor contacts.

ORDERING INFORMATION

t_{PD} (ns)	t_{EO} (ns)	f_{max} (MHz)	Order Code	Package	Operating Range
7.5	4.5	133.3	N85C508-7	PLCC	Commercial
			*D85C508-7	CERDIP	
			P85C508-7	PDIP	
10	6	100	N85C508-10	PLCC	Commercial
			*D85C508-10	CERDIP	
			P85C508-10	PDIP	

*Windowed package allows UV erase.

*ABEL is a trademark of Data I/O, Corp. CUPL is a trademark of Logical Devices, Inc. PLDesigner is a trademark of MINC, Inc.

ABSOLUTE MAXIMUM RATINGS*

Supply Voltage (V_{CC})(1) ..... −2.0V to +7.0V
Programming Supply Voltage (V_{PP})(1) ..... −2.0V to +13.5V
D.C. Input Voltage (V_I)(1, 2) ... −0.5V to V_{CC} + 0.5V
Storage Temperature (T_{stg}) ..... −65°C to +150°C
Ambient Temperature (T_{amb})(3) ... −10°C to +85°C

NOTES:
1. Voltages with respect to GND.
2. Minimum D.C. input is −0.5V. During transitions, the inputs may undershoot to −2.0V or overshoot to +7.0V for periods of less than 20 ns under no load conditions.
3. Under bias. Extended Temperature versions are also available.

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

2

RECOMMENDED OPERATING CONDITIONS

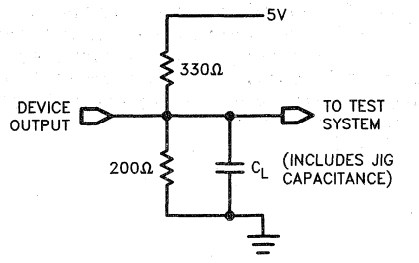
Symbol	Parameter	Min	Max	Units
V_{CC}	Supply Voltage	4.75	5.25	V
V_{IN}	Input Voltage	0	V_{CC}	V
V_O	Output Voltage	0	V_{CC}	V
T_A	Operating Temperature	0	+70	°C
t_R	Input Rise Time		500	ns
t_F	Input Fall Time		500	ns

D.C. CHARACTERISTICS (T_A = 0°C to +70°C, V_{CC} = 5.0V ±5%)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
V_{IH}(4)	High Level Input Voltage		2.0		V_{CC} + 0.3	V
V_{IL}(4)	Low Level Input Voltage		−0.3		0.8	V
V_{OH}	High Level Output Voltage	I_O = −4.0 mA D.C., V_{CC} = min	2.4			V
V_{OL}(5)	Low Level Output Voltage	I_O = 12.0 mA D.C., V_{CC} = min			0.45	V
I_I	Input Leakage Current	V_{CC} = max., GND < V_{IN} < V_{CC}			±10	µA
I_{SC}(6)	Output Short Circuit Current	V_{CC} = max., V_{OUT} = 0.5V	−30		−120	mA
I_{SB}(7)	Standby (Quiescent) Current	V_{CC} = max., V_{IN} = V_{CC} or GND, No Load, f_{IN} = 0 MHz			10	µA
I_{CC}	Power Supply Current	V_{CC} = max., V_{IN} = V_{CC} or GND, No Load, f_{IN} = 50 MHz, Device Prog. as 16-Bit Address Decoder		15	48	mA

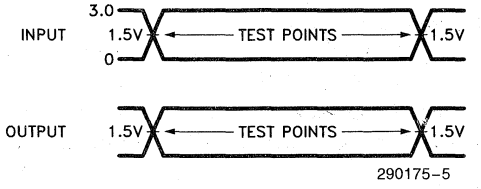
NOTES:
4. Absolute values with respect to device GND; all over and undershoots due to system or tester noise are included. Do not attempt to test these values without suitable equipment.
5. Maximum DC I_{OL} for the device (all outputs) is 64 mA.
6. Not more than 1 output should be tested at a time. Duration of that test should not exceed 1 second.
7. Standby current is higher when true and complement p-terms for the same input are both programmed.

A.C. TESTING LOAD CIRCUIT



C_L = 30 pF

A.C. TESTING INPUT, OUTPUT WAVEFORM



A.C. Testing: Inputs are driven at 3.0V for a Logic "1" and 0V for a Logic "0". Timing Measurements are made at 1.5V for inputs and outputs. Device input rise and fall times are less than 3 ns.

CAPACITANCE T_A = 0°C to +70°C; V_{CC} = 5.0V ± 5%

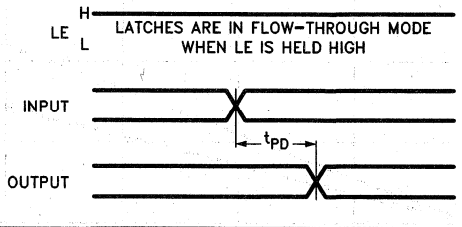
Symbol	Parameter	Conditions	Min	Typ	Max	Units
C_{IN}	Input Capacitance	V_{IN} = 0V, f = 1.0 MHz		6	10	pF
C_{OUT}	Output Capacitance	V_{OUT} = 0V, f = 1.0 MHz		6	10	pF
C_{CLK}	LE Capacitance	V_{IN} = 0V, f = 1.0 MHz		6	10	pF
C_{VPP}	V_{PP} Pin Capacitance	V_{PP} on Pin 1, f = 1.0 MHz		20	40	pF

A.C. CHARACTERISTICS T_A = 0°C to +70°C, V_{CC} = 5.0V ± 5%

Symbol	Parameter	85C508-7			85C508-10			Units
		Min	Typ	Max	Min	Typ	Max	
t_{PD}(8)	Propagation Delay (Flow-Through Mode)	3	6	7.5	3	8	10	ns
f_{max}	Maximum Frequency ($1/t_{CW}$)		166	133.3		112	100	MHz
t_{EO}(8)	Output Valid from LE ↑	0.5	4	4.5	0.5	5	6	ns
t_{SU}	Input Setup Time to LE ↓	5.5	4		7	5		ns
t_H	Input Hold from LE ↓	−2	−3		−3	−5		ns
t_{CH}	LE High Time	4			5			ns
t_{CW}	LE ↑ to LE ↑	7.5			10			ns

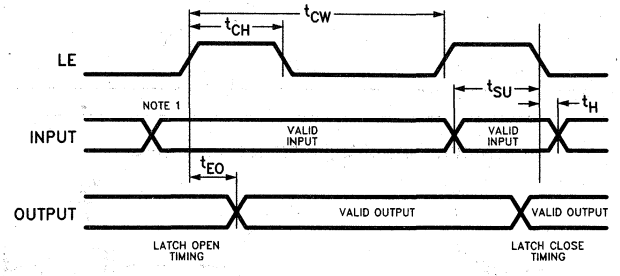
NOTE:
8. One output going active; one output going inactive.

FLOW-THROUGH MODE



290175-6

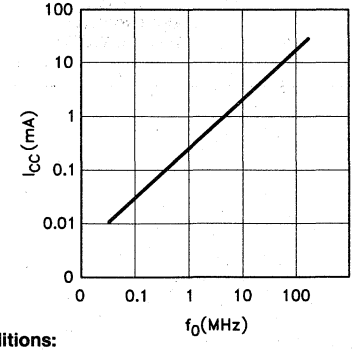
LATCH MODE



290175-7

NOTE:
1. t_{EO} assumes a system setup min. of (t_{PD}–t_{EO}); e.g. 3 ns setup for a 85C508-7.

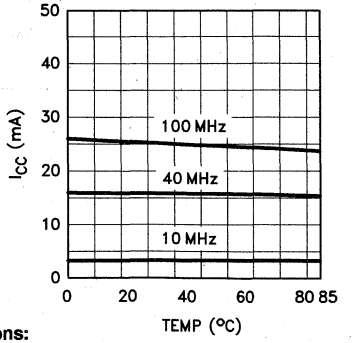
85C508 I_{CC} vs Frequency



Conditions:
T_A = 25°C
V_{CC} = 5.0V
C_L = 30 pF

290175-9

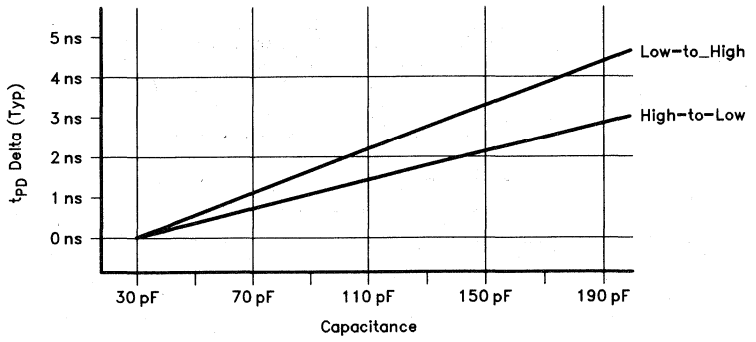
85C508 I_{CC} vs Temperature



Conditions:
V_{CC} = 5.0V
C_L = 30 pF

290175-10

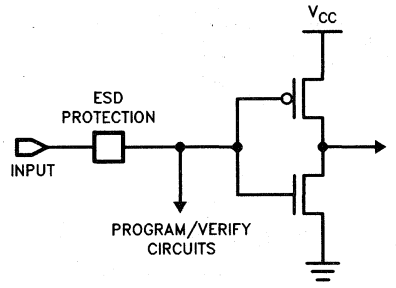
85C508 t_{PD} Derating vs Capacitive Loading



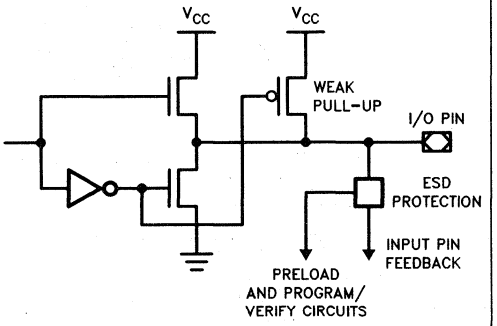
Conditions:
T_A = 25°C
V_{CC} = 5.0V

290175-11

Input/Output Equivalent Schematics



290175-12



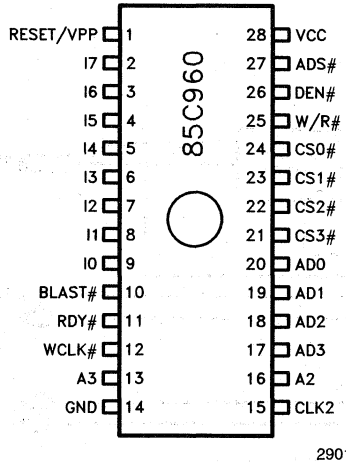
290175-13

85C960
1-MICRON CHMOS
80960 K-SERIES BUS CONTROL μPLD

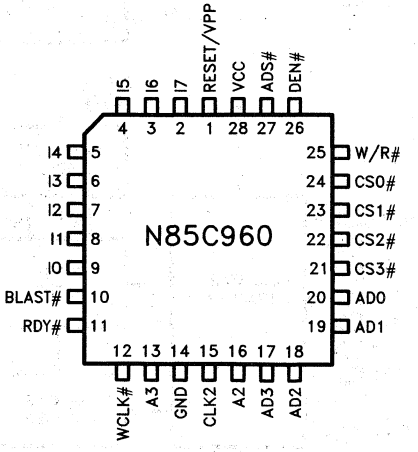
- **Burst Logic, Ready Control, and Address Decode Support for 80960 KA/KB Embedded Controllers in Single Chip**
- **Burst Logic Supports Both Standard and New Generation "Burst Mode" Memories and Peripherals**
- **Ready/Timing Control Supports 0–15 Wait States across 8 Address Ranges, Read/Write Accesses, Burst Transactions**
- **8 Dedicated Inputs Decoded into 8 Latched Chip Selects (4 External/ Internal; 4 Internal Only)**
- **Operates with 80960KA/KB at 20 MHz and 25 MHz**
- **I_{CC} = 50 mA Max.**
- **UV Erasable (CerDIP) or OTP™**
- **100% Generically Testable Logic Array**
- **Based on Low Power CHMOS IIIE* Technology**
- **Available in 28-Pin 300-mil CerDIP and PDIP Packages and in 28-Pin PLCC Package**
  (See Packaging Spec., Order Number #231369)

*CHMOS is a patented technology of Intel Corporation.

2



290192-1



290192-2

= **Active Low Signals**

Figure 1. Pinout Diagram

August 1990
Order Number: 290192-002

GENERAL DESCRIPTION

The Intel 85C960 is a single-chip burst/ready/decode μPLD (Microcomputer Programmable Logic Device) designed to interface 80960 KA/KB embedded controllers to system memory and I/O. The 85C960 provides programmable chip selects, a programmable read/write access wait state/ready generator, and burst address (A2, A3) cycling. Burst transaction cycling of A2, A3, and WCLK# (Write Clock) is also supported for intelligent peripherals on the bus.

For its programmable functions, the 85C960 uses advanced EPROM cells as logic array and wait-state table memory elements. Coupled with Intel's proprietary CHMOS IIIE technology, the result is a programmable device able to support Intel's 32-bit 80960 KA/KB embedded controllers at speeds up to 25 MHz.

ARCHITECTURE DESCRIPTION

The 85C960 μPLD integrates burst control, ready generation, and chip select decoding into a single device. Figure 2 shows the architecture of the 85C960. Table 1 lists and describes each signal on the device. The 85C960 replaces 6–10 separate PLD/discrete logic devices in small- and medium-sized 80960 systems. For medium- to large-sized systems, the 85C960 can be supplemented with an additional decoder, such as the 85C508, and a second 85C960. Figure 3 shows a single 85C960 in a typical application.

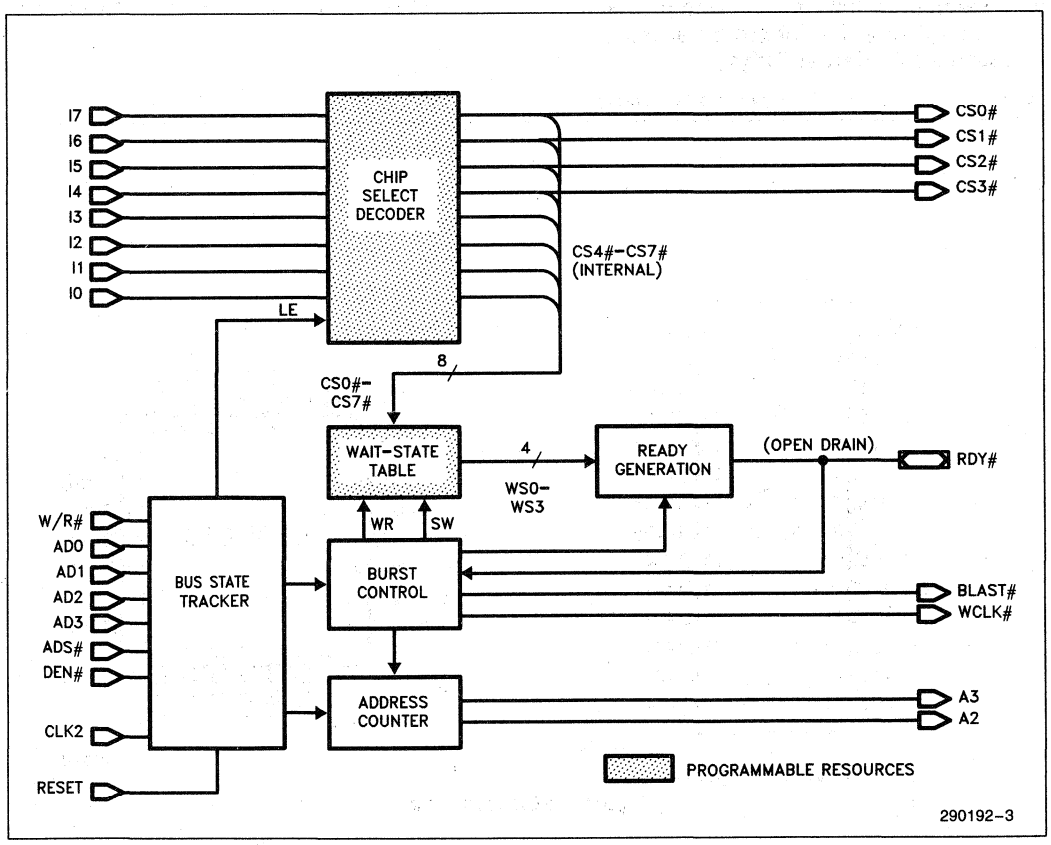


Figure 2. 85C960 Block Diagram

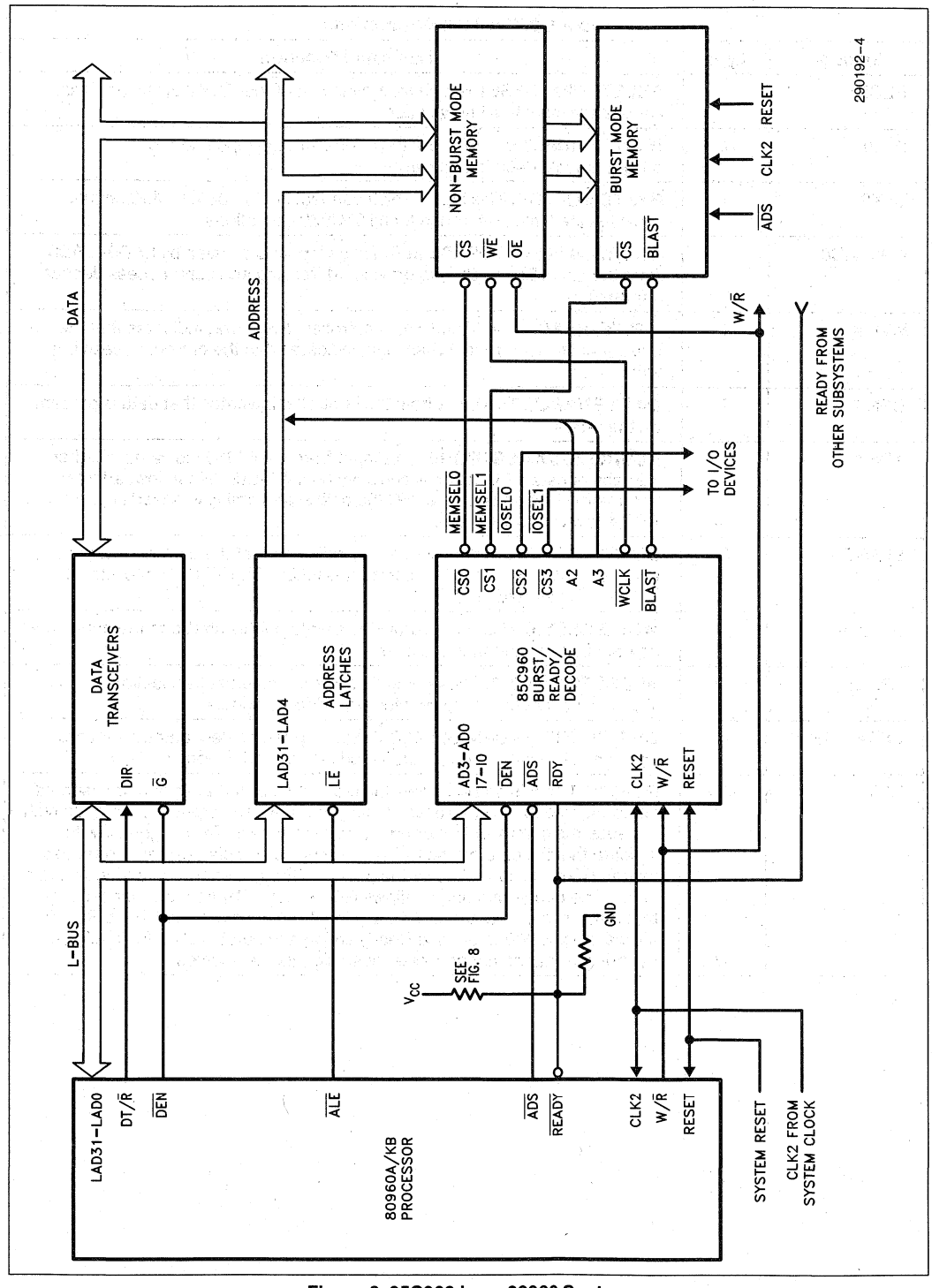


Figure 3. 85C960 in an 80960 System

Table 1. 85C960 Pin Descriptions

Symbol	Type	Name and Function
RESET	I	**RESET.** When RESET is high for a minimum of four CLK2 cycles, internal circuits are reset to a known state.
I7–I0	I	**INPUT 7–INPUT 0.** These are the address range inputs to the programmable decode logic array.
CLK2	I	**SYSTEM CLOCK.** This input, which connects to the 80960 CLK2 signal, provides the timing reference for all 85C960 operations.
AD3–AD0	I	**ADDRESS IN 3–ADDRESS IN 0.** These inputs are driven by LAD0–LAD3 from the Local Bus (L-Bus) to provide addressing and burst access decode information.
W/R#	I	**WRITE/READ.** Write/Read from controller. When low, indicates that the current access is a read. When high, indicates that the current access is a write.
DEN#	I	**DATA ENABLE.** This input from the controller indicates that data is present on the L-Bus.
ADS#	I	**ADDRESS/DATA STROBE.** This input from the 80960 indicates whether address or data information is currently on the L-Bus. When low, address information is changing. The 85C960 chip select timing is based in part on ADS# low during Ta states.
BLAST#	O	**BURST LAST.** This signal, when low, indicates that the current read/write access is the last access in a burst transaction. BLAST# is not cycled if RDY# is generated off-chip.
WCLK#	O	**WRITE CLOCK.** This output provides a write enable strobe to memories that do not support burst mode access.
A3, A2	O	**ADDRESS OUT 3, 2.** These outputs cycle during burst transactions. Typically connected to lowest memory address signals.
CS3#–CS0#	O	**CHIP SELECT 3–CHIP SELECT 0.** Single p-term select outputs that are driven active (low) for the programmed address condition on I7–I0.
RDY#	I/O	**READY.** RDY# is an active low, bidirectional, open-drain signal that should be connected to the controller's Ready input. As an output, RDY# goes high to cause the controller to extend the current access. RDY# goes low to indicate that the data on the L-Bus bus may be sampled (read) or removed (write). RDY# is controlled by the 85C960 Ready Generation and Wait-State Logic. The open-drain output allows RDY# to be OR-tied to other circuitry that may drive the controller's Ready input. As a bidirectional input, RDY# allows the 85C960 to provide Ready timing and burst cycling for intelligent peripherals that do not generate these signals themselves.

80960 L-Bus (Local Bus) cycles are monitored by the **Bus State Tracker** to synchronize the functional blocks in the 85C960 to the L-Bus. CLK2 provides the timing reference for all 85C960 operations.

Four external chip selects (CS0#–CS3#) are generated by the programmable **Chip Select Decoder.** These four signals provide decoded selects to memory and I/O devices and are routed to the programmable **Wait-State Table** so that the 85C960 can generate RDY# at the appropriate time. Four additional selects are decoded (internal only) and routed to the Wait-State Table so that the 85C960 can generate RDY# for up to four additional address ranges.

The **Ready Generation** block generates RDY# to the controller under control of the **Wait-State Table.** Depending on the contents programmed into this table and the current type of access, from 0–15 wait states can be introduced into each bus cycle. An independent wait state value can be chosen for each select and each access type. Four access types are possible: read first, read subsequent, write first, and write subsequent.

The **Burst Control** and **Address Counter** blocks control burst transaction timing to memory and I/O. Note that the RDY# pin is sampled by the Burst Control block to allow the 85C960 to generate burst transaction timing for other bus peripherals. WCLK# provides a write enable strobe for memory and I/O that do not support burst mode. BLAST# informs burst-mode devices that the current access is the last one in a burst transaction. A2 and A3 are cycled to select the address location for each access.

FUNCTIONAL DESCRIPTION

The following paragraphs provide a detailed description of each functional block in the 85C960 μPLD.

Chip Select Decoder

The Chip Select Decoder, shown in Figure 4, is a high speed, single p-term (product-term) latched decoder circuit with eight inputs (I0–I7) and eight latched outputs. Each output goes low when its associated product term is true. Four of these outputs (CS0#–CS3#) are available externally to be used as device selects. The remaining four outputs (CS4#–CS7#) are available internally so that the 85C960 can provide ready and burst timing for four more device selects. (The actual selects for these four additional devices/resources must be generated by external logic.)

The input to each latch is a single NAND p-term that can be connected to the dedicated inputs. The true and complements of all inputs (I7–I0) are available to all eight NAND p-terms.

Each intersecting point in the logic array is connected or not connected based on the value programmed in the EPROM array. Initially (EPROM erased state), no connections exist between any p-term and any input. Connections can be made by programming the appropriate EPROM cells. Since p-terms are implemented as NANDs, a true condition on a p-term drives the output low. Current consumption is higher when both true and complement p-terms for the same input are programmed.

Selects are latched on the falling edge of an internal Latch Enable (LE), which is generated from ADS#, DEN#, and CLK2. The proper combination of these signals occurs during an 80960 address state (Ta). Figure 5 shows the relationship of the internal LE and external chip selects to the three signals at the end of a Ta state. All selects are cleared to an inactive high state at the start of a recovery state. (Tr). All eight selects (four external and four internal) are routed to the Wait-State Table.

Wait State Table

Chip selects, WR (Write/Read), and SW (Subsequent Word) feed the Wait-State Table. Each chip select points to a set of four wait state values while WR and SW determine which of the four values to route to the Ready Generation block (see Figure 6). The four values are grouped into read and write groups with each group having a value for the first access and subsequent access (second through fourth). The four-bit wait-state value is sent to the Ready Generation block (via WS0#–WS3#) to be used as an initial count value. If two selects are active, the resulting count value is the logical bit AND of the two individual values. If more than two selects are active and the individual count values are not the same, the resulting count value is indeterminate. If no select is active, no count value is loaded (and the Ready Generation circuit is disabled).

Ready Generation

RDY# is high at the start of each burst transaction. The RDY Generator begins to count down from the wait state value, decrementing the counter at the start of each wait state. When the internal counter reaches 0000, RDY# is pulled low (CLK2c during the data state). On the next CLK2c edge (for a wait state), RDY# is released, allowing an external resistor to pull RDY# high. Figure 7 shows the timing for a four-word burst write transaction with 1 wait state for the first access and 0 wait states for the remaining three accesses (Burst Write 1-0-0-0).

RDY# is an open-drain I/O pin, which must be connected to pullup and pulldown resistors as shown in Figure 8. During a wait-state access, RDY# is pulled high to cause the controller to extend the current access so that the memory or peripheral chip has time to present data to the bus (read), or sample data on the bus (write). RDY# is released on the CLK2a edge of a Tr state. If a Read or Write access occurs without a chip select having been decoded on-chip, the RDY# output buffer is disabled and RDY# is sampled as an input. This allows the 85C960 to cycle A2, A3, and WCLK# to provide burst transaction timing for other bus controllers. RDY# may be OR-tied with other bus controllers so they can access the processor Ready signal.

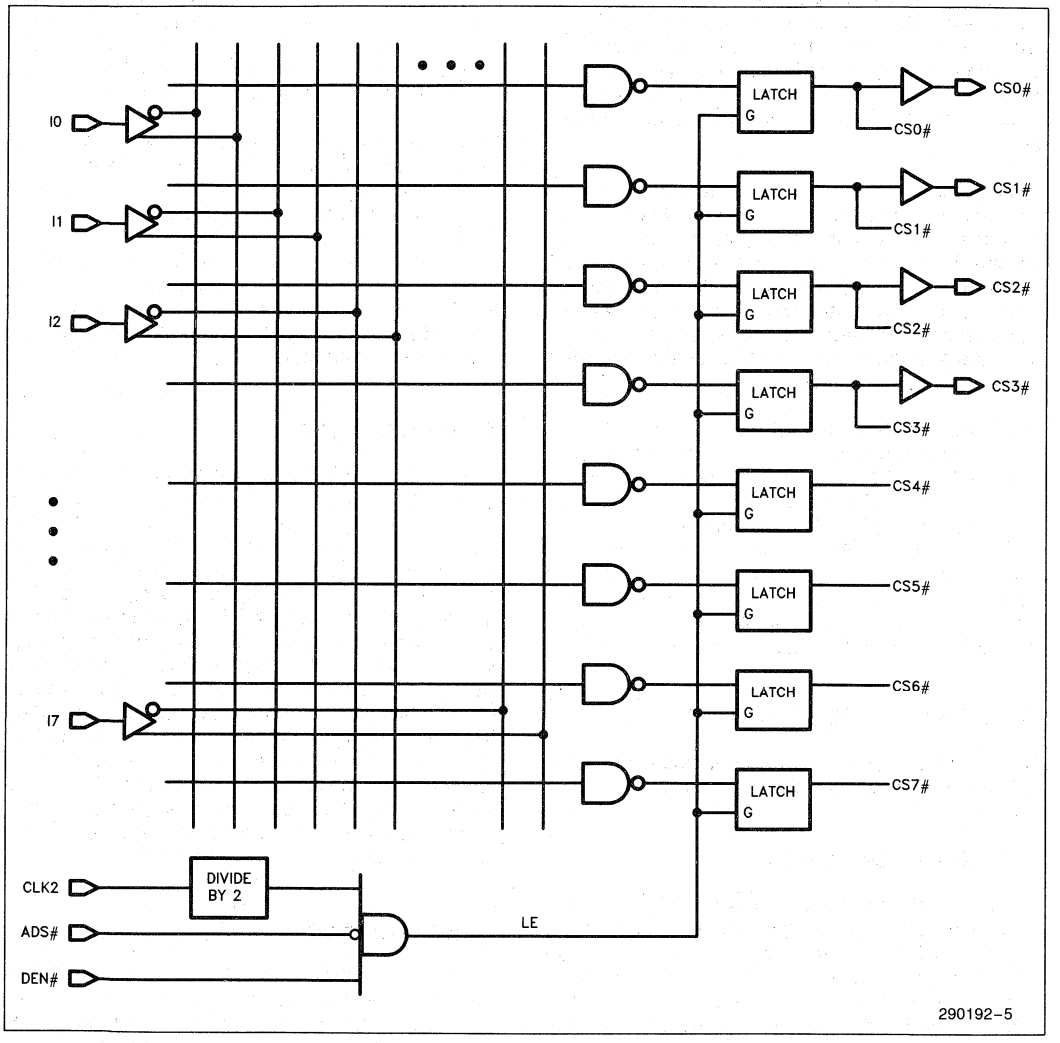
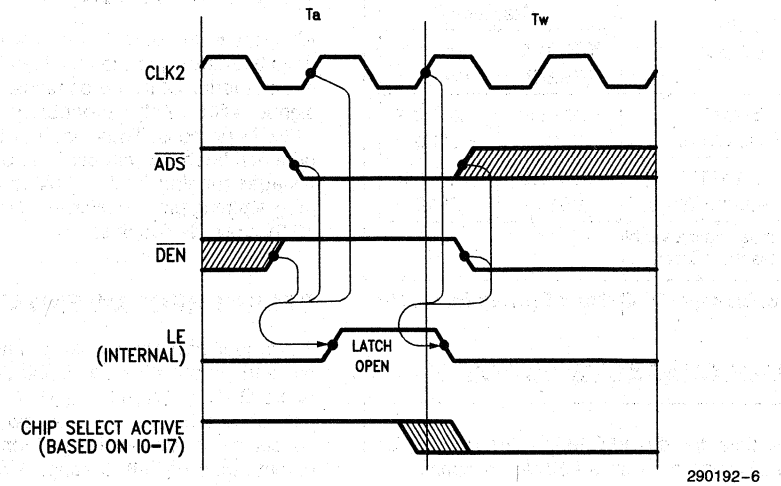


Figure 4. 85C960 Chip Select Decoder Block



Latch opens when CLK2 and DEN# go high and ADS# goes low.
Latch closes when DEN# goes low or ADS# or CLK2 go high.

Figure 5. Internal LE and External Chip Select Timing

Burst Transactions

AD3, AD2 are latched to indicate the starting address of a burst transaction. The 85C960 places these two signals out on A3 and A2, respectively, then cycles the two addresses upward until the last access of the burst. The 85C960 assumes that the processor handles splitting of the burst transaction when a 16-byte boundary is crossed.

AD0 and AD1 specify the size of the burst transfer in double-words as shown in Table 2.

Table 2. AD0–AD1 vs Burst Size

AD1	AD0	No. of Words Transferred
0	0	1
0	1	2
1	0	3
1	1	4

WCLK#, BLAST# Generation

WCLK# is the write enable signal for writing to non-burst mode memories. When low, address outputs A2 and A3 are valid. Its trailing edge (low-to-high transition) can be used to latch data into non-burst mode memories. WCLK# is only provided during writes; during reads, WCLK# remains high.

BLAST# indicates that the current access is the last access in a burst transaction. BLAST# is used by burst-mode memories to reset internal address counters. BLAST# is not cycled when RDY# is generated off-chip.

POWER-ON CHARACTERISTICS

85C960 inputs and outputs begin responding 1 μs (max.) after V_{CC} power-up (V_{CC} = 4.75V) or after a power-loss/power-up sequence. RESET must be synchronous to CLK2 and must be held high for a minimum of 4 clock cycles after V_{CC} reaches 4.75 V. After 4 clock cycles, A2 and A3 are high, CS0# – CS3# (and CS4# – CS7#), BLAST#, WCLK# are high, and the open drain RDY# signal is inactive.

Select CS0f#	Write/Read	
	WR = 0 (Read)	WR = 1 (Write)
SW = 0 (First Word)	msb lsb 0000	msb lsb 0000
SW = 1 (Subsequent Word)	msb lsb 0011	msb lsb 0010

msb = most significant bit
lsb = least significant bit

Figure 6. Example Wait-State Entries for CS0f#

ERASURE CHARACTERISTICS

Erasure time for the 85C960 is 20 minutes at 12,000 μWsec/cm^2 with a 2537Å UV lamp.

Erasure characteristics of the device are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000Å. It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000Å–4000Å range. Data shows that constant exposure to room level fluorescent lighting could erase the typical 85C960 in approximately two years, while it would take approximately two weeks to erase the device when exposed to direct sunlight. If the device is to be exposed to these lighting conditions for extended periods of time, conductive opaque labels should be placed over the device window to prevent unintentional erasure.

The recommended erasure procedure for the 85C960 is exposure to shortwave ultraviolet light with a wavelength of 2537Å. The integrated dose (i.e., UV intensity $\times$ exposure time) for erasure should be a minimum of fifteen (15) Wsec/cm^2. The erasure time with this dosage is approximately 20 minutes using an ultraviolet lamp with a 12,000 μW/cm^2 power rating. The device should be placed within 1 inch of the lamp tubes during exposure. The maximum integrated dose the 85C960 can be exposed to without damage is 7258 Wsec/cm^2 (1 week at 12,000 μW/cm^2). Exposure to high intensity UV light for longer periods may cause permanent damage to the device.

LATCH-UP IMMUNITY

All of the input, output, and clock pins of the device have been designed to resist latch-up which is inherent in inferior CMOS processes. The 85C960 is designed with Intel's proprietary 1-micron CHMOS EPROM process. Thus, each of the pins will not experience latch-up with currents up to $\pm$100 mA and voltages ranging from -0.5V to (V_{CC} + 0.5V). The programming pin is designed to resist latch-up to the 13.5V max. device limit.

DESIGN RECOMMENDATIONS

For proper operation, it is recommended that all input and output pins be constrained to the voltage range GND $\leq$ (V_{IN} or V_{OUT}) $\leq$ V_{CC}. All unused inputs should be tied high or low to minimize power consumption (do not leave them floating). Unused outputs may be left floating. A high-speed ceramic decoupling capacitor of at least 0.2 μF must be connected directly between the V_{CC} and GND pin.

As with all CMOS devices, ESD handling procedures should be used with the 85C960 to prevent damage to the device during programming, assembly, and test.

FUNCTIONAL TESTING

Since the programmable sections of the 85C960 are controlled by EPROM elements, the device is completely testable during the manufacturing process. Each programmable EPROM bit controlling the internal logic is tested using application independent test patterns. EPROM cells in the device are 100% tested for programming and erasure. After testing, the devices are erased before shipments to the customers. No post-programming tests of the EPROM array are required.

The testability and reliability of EPROM-based programmable logic devices is an important feature over similar devices based on fuse technology. Fuse-based programmable logic devices require a user to perform post-programming tests to insure device functionality. During the manufacturing process, tests on fuse-based parts can only be performed in very restricted ways in order to avoid preprogramming the array.

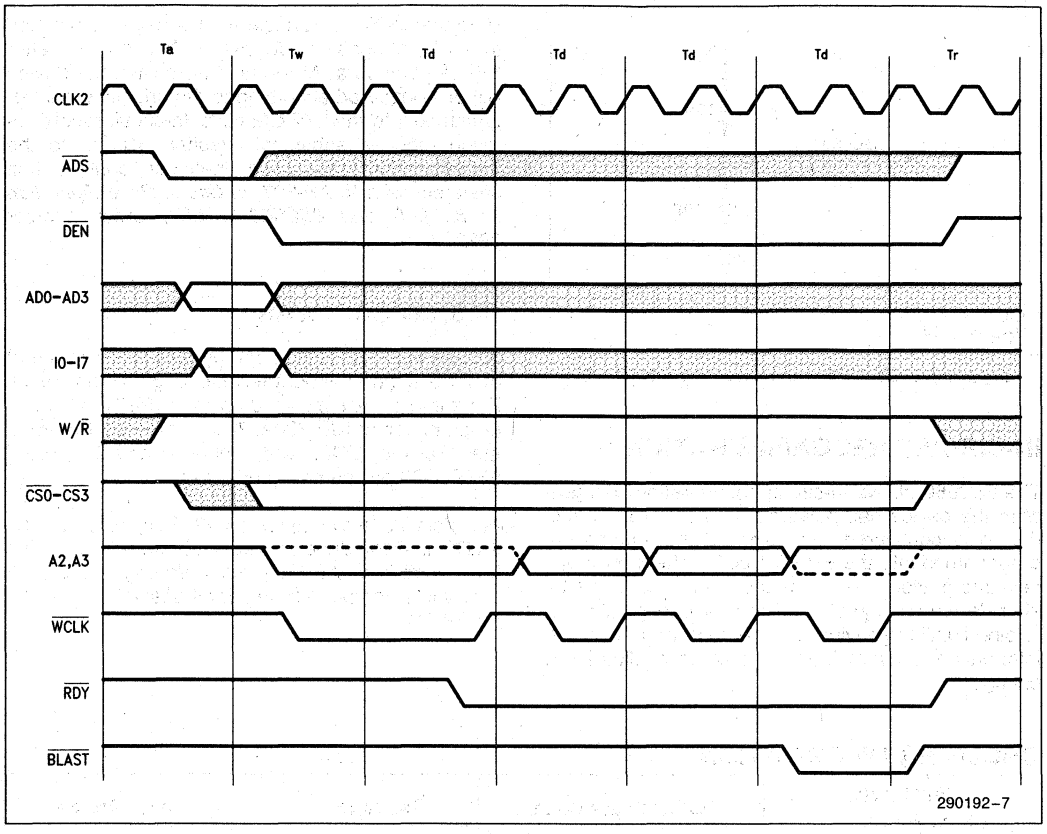


Figure 7. Burst Write Transaction (1-0-0-0)

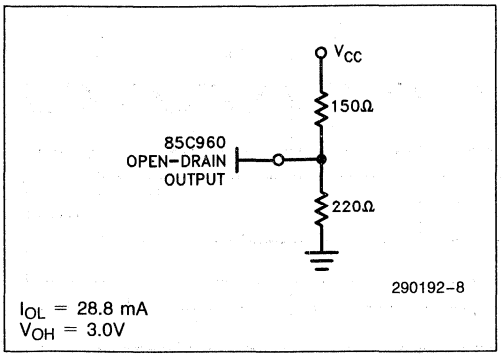


Figure 8. RDY# Pullup/Pulldown Resistors

IN-CIRCUIT RECONFIGURATION

The 85C960 allows in-circuit configuration changes after the device has powered up. At power-up, the device is configured according to the information programmed into the EPROM cells. After power-up, new information can be shifted in on select pins to alter device configuration. The new configuration is retained until the device is powered down or until the information is overwritten by another configuration change.

Note that in-circuit configuration changes allow "on-the-fly" changes to be made, but do not alter EPROM cell data. At the next power-up, the device will be configured according to the original data programmed into the EPROM cells. In-circuit reconfiguration requires additional circuitry external to the 85C960. For details on in-circuit configuration changes, refer to AP-337, *In-Circuit Reconfiguration of 85C960 and 85C508 μPLDs,* order number 292072.

DESIGN SOFTWARE

Software support is provided by version 2.1 (or later) of iPLS II (Intel Programmable Logic Software II). Programming is supported on the iUP-PC PC-based programmer or iUP-200A/201A Universal Programmer via the GUPI base module and the GUPI 85EPLD28 programming adaptor.

For detailed information on iPLS II, refer to the iPLDS II Data Sheet, order number: 290134. The tools section of the *Programmable Logic* handbook contains a complete listing of all design tools for Intel EPLDs.

ORDERING INFORMATION

80960KA/KB Clock Frequency	μPLD Order Code	Package	Operating Range
20 MHz	*D85C960-20	CERDIP	Commercial
	N85C960-20	PLCC	
25 MHz	*D85C960-25	CERDIP	Commercial
	N85C960-25	PLCC	

*Only windowed CERDIP allows UV-erase.

ABSOLUTE MAXIMUM RATINGS*

Supply Voltage (V_{CC})(1) . . . . . −2.0V to +7.0V

Programming Supply Voltage (V_{PP})(1) . . . . . −2.0V to +13.5V

D.C. Input Voltage (V_I)(1, 2) . . . −0.5V to V_{CC} + 0.5V

Storage Temperature (T_{stg}) . . . . . −65°C to +150°C

Ambient Temperature (T_A)(3) . . . . . −10°C to +85°C

NOTES:
1. Voltages with respect to GND.
2. Minimum D.C. input is −0.5V. During transitions, the inputs may undershoot to −2.0V or overshoot to +7.0V for periods of less than 20 ns under no load conditions.
3. Under bias. Extended Temperature versions are also available.

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

2

RECOMMENDED OPERATING CONDITIONS

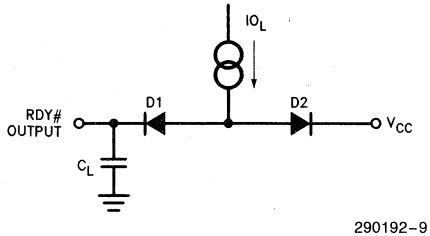
Symbol	Parameter	Min	Max	Units
V_{CC}	Supply Voltage	4.75	5.25	V
V_{IN}	Input Voltage	0	V_{CC}	V
V_O	Output Voltage	0	V_{CC}	V
T_A	Operating Temperature	0	+70	°C

D.C. CHARACTERISTICS (T_A = 0°C to +70°C, V_{CC} = 5.0V ± 5%)

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
V_{IH1}(4)	High Level Input Voltage (All Inputs except for ADS#, AD0–AD3, DEN#, and W/R#)	2.0		V_{CC} + 0.3	V	
V_{IH2}(4)	High Level Input Voltage for ADS#, AD0–AD3, DEN#, and W/R#	2.2			V	
V_{IL}(4)	Low Level Input Voltage	−0.3		0.8	V	
V_{OH}	High Level Output Voltage	2.4			V	I_{OH} = −4.0 mA D.C., V_{CC} = Min.
V_{OL1}	Low Level Output Voltage			0.4	V	I_{OL} = 4.0 mA D.C., V_{CC} = Min., C_L = 30 pF
V_{OL2}	Low Level Output Voltage for A2, A3			0.45	V	I_{OL} = 24 mA D.C., V_{CC} = Min., C_L = 60 pF
V_{OL3}	Low Level Output Voltage for Open Drain (RDY#)			0.5	V	I_{OL} = 30 mA D.C., V_{CC} = Min., C_L = 30 pF
I_I	Input Leakage Current			±10	μA	V_{CC} = Max., GND ≤ V_{IN} ≤ V_{CC}
I_{OZ}	Output Leakage Current			±10	μA	V_{CC} = Max., GND ≤ V_{OUT} ≤ V_{CC}
I_{SC}(5)	Output Short Circuit Current	−30		−90	mA	V_{CC} = Max., V_{OUT} = 0.5V
I_{CC}	Power Supply Current		10	50	mA	V_{CC} = Max., V_{IN} = V_{CC} or GND, No Load, CLK2 = 50 MHz

NOTES:
4. Absolute values with respect to device GND; all over and undershoots due to system or tester noise are included.
5. Not more than 1 output should be tested at a time. Duration of that test should not exceed 1 second.

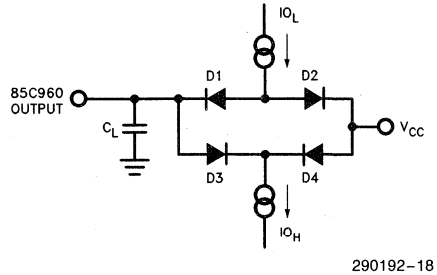
A.C. TESTING LOAD CIRCUIT (RDY#)



290192-9

See D.C. Characteristics Table for Current and Capacitance Specifications.
D1 and D2 are matched.

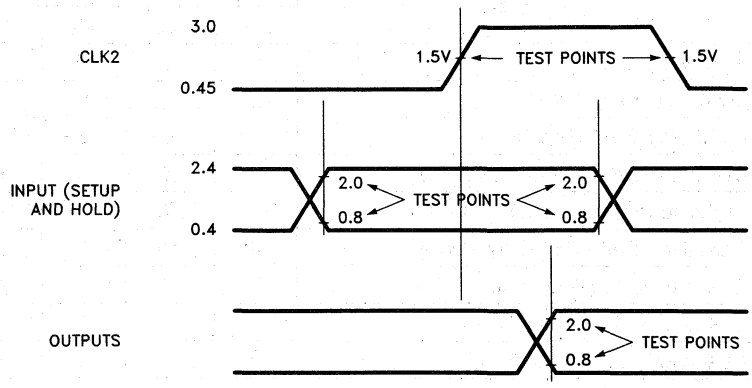
A.C. TESTING LOAD CIRCUIT (ALL OUTPUTS EXCEPT RDY#)



290192-18

See D.C. Characteristics Table for Current and Capacitance Specifications.
D1 and D2 are matched
D3 and D4 are matched

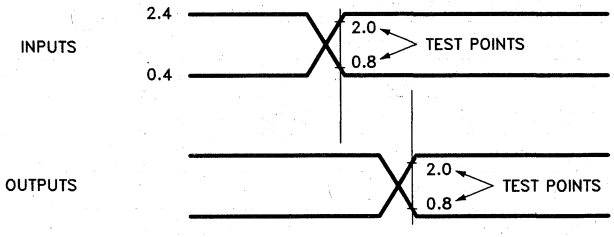
A.C. TESTING WAVEFORM—SYNCHRONOUS INPUTS AND OUTPUTS



290192-10

A.C. Testing: Inputs are driven at 2.4V for a Logic "1" and 0.4V for a Logic "0". CLK2 is driven at 3.0V for a Logic "1" and 0.45V for a Logic "0". Timing Measurements made relative to CLK2 are made from 1.5V on CLK2. Inputs and outputs are measured at 2.0V for a high and 0.8V for a low. Device input rise and fall times are less than 3 ns.

A.C. TESTING WAVEFORM—ASYNCHRONOUS INPUTS AND OUTPUTS



290192-11

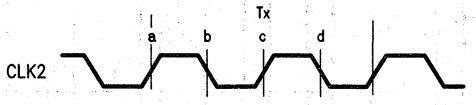
A.C. Testing: Inputs are driven at 2.4V for a Logic "1" and 0.4V for a Logic "0". Input timing is measured at 1.5V for high-to-low and low-to-high transitions. Outputs are measured at 2.0V for a high and 0.8V for a low. Device input rise and fall times are less than 3 ns.

A.C. CHARACTERISTICS (T_A = 0°C to +70°C, V_{CC} = 5.0V ±5%)

Symbol	Parameter	85C960-25		85C960-20		Units
		Min	Max	Min	Max	
t_1(6)	Input Setup to CLK2a	12		15		ns
t_2(6)	Input Hold from CLK2a	2		2		ns
t_3	CLK2a to A2, A3 Valid Delay	0	8	0	10	ns
t_4	CLK2c to RDY# Output Low Delay		10		15	ns
t_5(7)	CLK2c to RDY# Output High Delay		10		15	ns
t_6	CLK2a to CS0#-CS3# High Delay	5	40	5	50	ns
t_7	CLK2a to BLAST# Low Delay		20		20	ns
t_8	CLK2a to BLAST# High Delay	5		5		ns
t_9(8)	CLK2b to WCLK# Low Delay	0	10	0	12	ns
t_{10}(8)	CLK2d to WCLK# High Delay	0	10	0	12	ns
t_{11}(9)	ADS# Low to CS0#-CS3# Low Delay		10		12	ns
t_{12}(9)	CLK2c to CS0#-CS3# Low Delay		12		15	ns
t_{13}(10)	I0-I7 Setup to CLK2a	5		7		ns
t_{14}(10)	I0-I7 Hold from CLK2a	2		2		ns
t_{15}(11)	I0-I7 Valid to CS0#-CS3# Valid Delay-(t_{PD})		10		12	ns
t_{16}	RDY# Input Setup to CLK2d (Write)	7.5		10		ns
t_{17}	RDY# Input Setup to CLK2a (Read)	9		9		ns
t_{18}	RDY# Input Hold after CLK2a (Read/Write)	5		10		ns
t_{19}(12)	RESET High Setup to CLK2 ↑	0		0		ns
t_{20}(13)	RESET High Hold from CLK2 ↑	3		3		ns
t_{21}(12)	RESET Low Setup to CLK2a	5		5		ns

NOTES:
6. Applies to ADS#, DEN#, W/R#, and AD0-AD3. DEN# is high during the entire Ta state in 80960 KA/KB systems.
7. RDY# is an open-drain output. Specified time includes RDY# output float delay and pull-up/pull-down resistors (Figure 8). RDY# remains low for a minimum of 10 ns at the start of a Tr state and goes high by CLK2a of the next Tx state.
8. Minimum WCLK# pulse width is one clock period minus 3 ns. For example, at 25 MHz: 20 ns − 3 ns = a 17 ns minimum WCLK# pulse.
9. Chip Select Decoder latches are transparent flow-through types. Latches open when ADS# is low, DEN# is high, and CLK2 goes high during the middle of a Tx state (CLK2c). Since DEN# is high during the entire Ta state in 80960 KA/KB systems, only CLK2c and ADS# are specified.
10. Chip Select Decoder latches are transparent flow-through types. Latches close when ADS# is high or DEN# is low, or when CLK2 goes high at the start of a Tx state (CLK2a) after the latches have opened. Since ADS# is low and DEN# is high at the end of a Ta in 80960 KA/KB systems, setup and hold times are specified with reference to CLK2a only.
11. Propagation delay while latches are open (transparent); one output switching (high-to-low).
12. RESET must be held high for a minimum of 4 CLK2 cycles (80960 specifies 41 CLK2 cycles minimum).
13. RESET must hold after the low-to-high transition immediately prior to CLK2a. CLK2a is defined as the first low-to-high transition after RESET goes low.

CLK2 EDGES



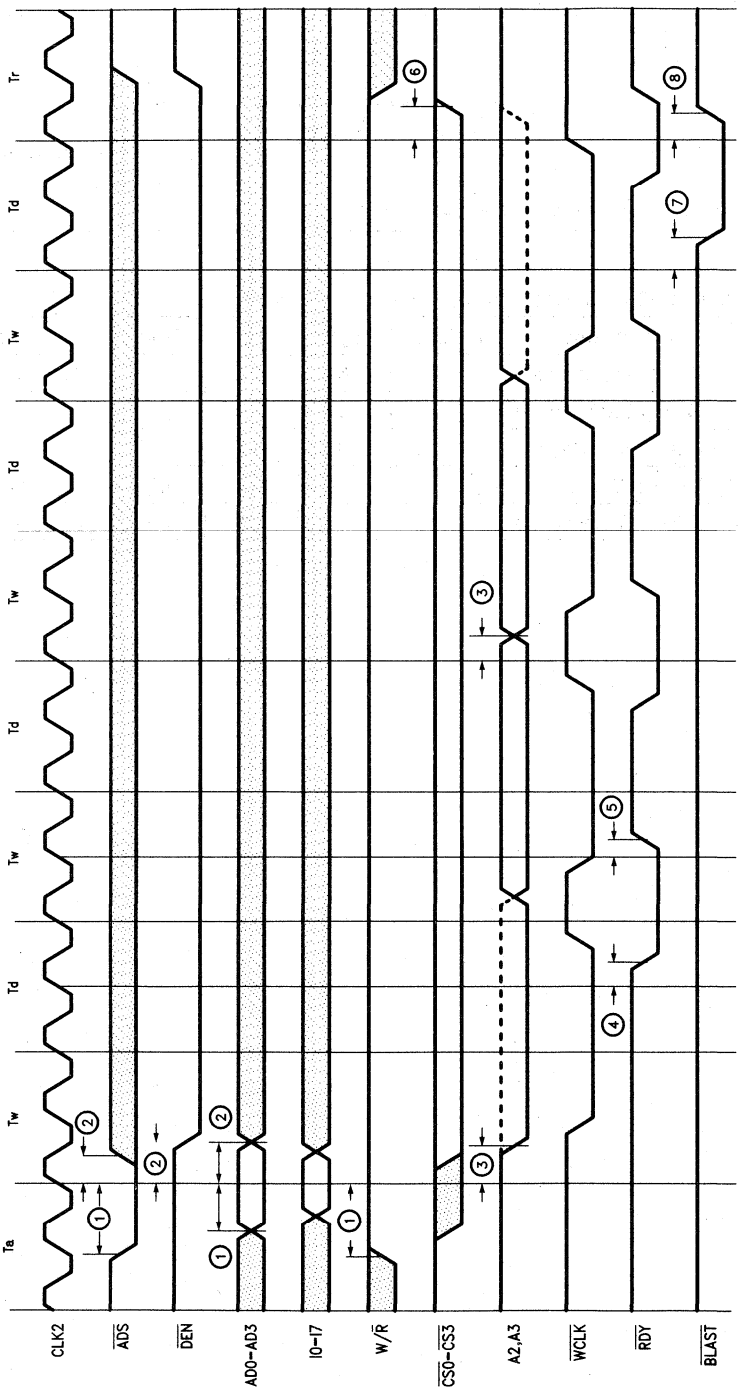
290192-12

NOTE:
Minimum CLK2 high and low times are 8 ns measured from 1.5V to 1.5V.

CAPACITANCE (T_A = 0°C to +70°C; V_{CC} = 5.0V ± 5%)

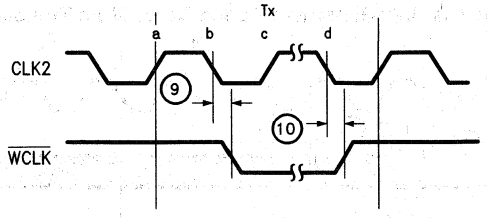
Symbol	Parameter	Min	Typ	Max	Unit	Conditions
C_{IN}	Input Capacitance		6	10	pF	V_{IN} = 0V, f = 1.0 MHz
C_{OUT}	Output Capacitance		6	10	pF	V_{OUT} = 0V, f = 1.0 MHz
C_{CLK}	CLK2 Capacitance		6	10	pF	V_{IN} = 0V, f = 1.0 MHz
C_{VPP}	V_{PP} Pin Capacitance		10	25	pF	V_{PP} on Pin 1 (RESET)
C_{RDY}	RDY# Capacitance		6	10	pF	V_{OUT} = 0V, f = 1.0 MHZ

4 Word Burst Write with 1 Wait State on Each Access
RDY # is Generated by the 85C960
(Same Timing for Read Cycle, Except WCLK # Remains High)



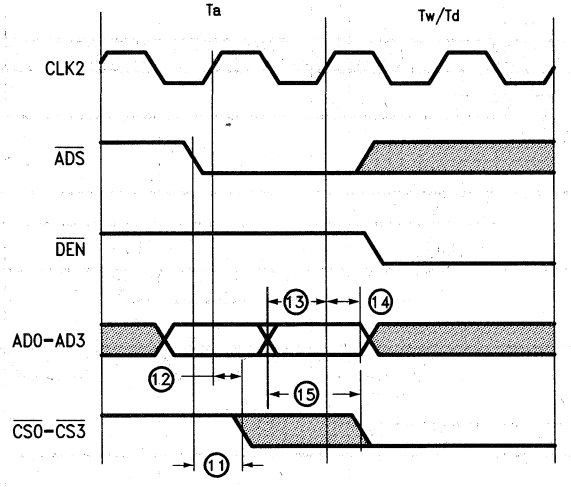
290192-13

WCLK# TIMING



290192-14

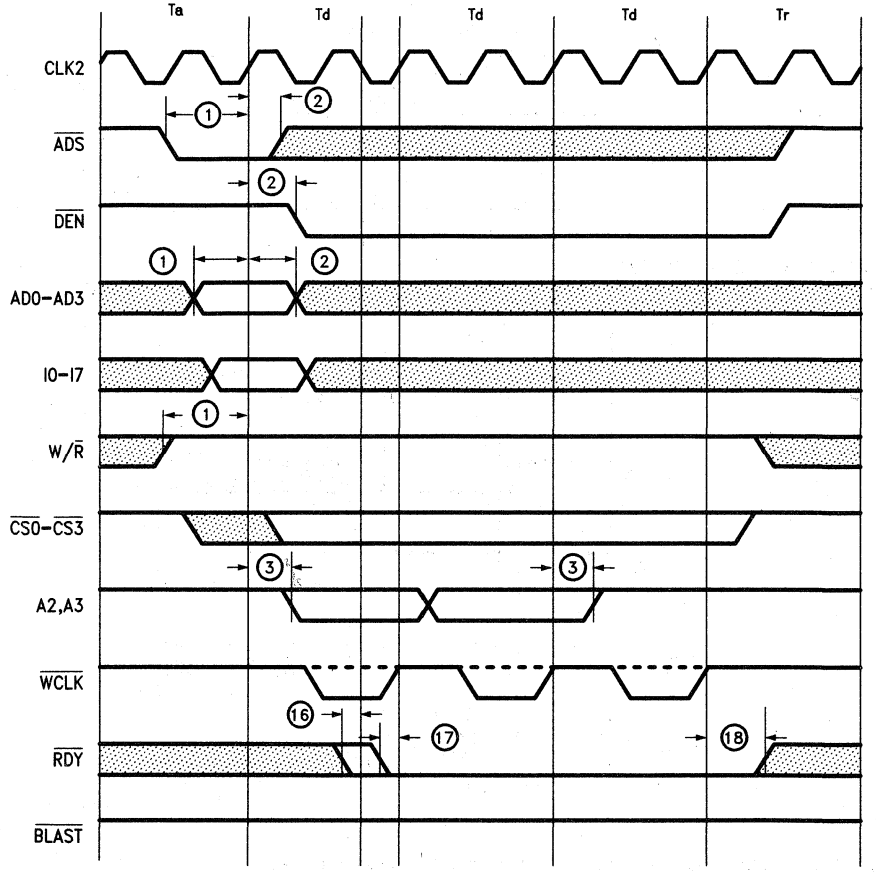
I0-I7 AND CS0#-CS3# TIMING



290192-15

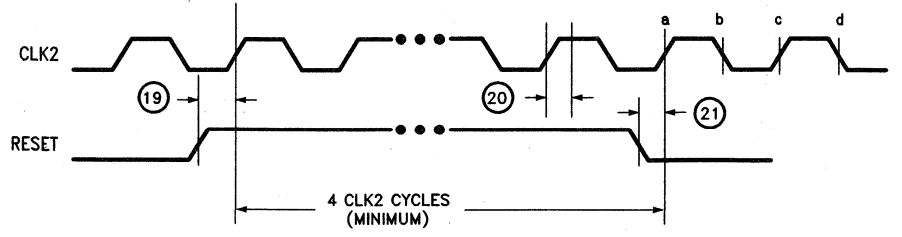
NOTE:
CLK2, ADS#, and DEN# generate internal latch enable. See Figure 7 for details.

3 Word Burst with 0 Wait States on Each Access
RDY# is Generated Externally
(WCLK# is Only Generated During Burst Write Transactions)



290192-16

RESET INPUT TIMING



290192-17

5AC312
1-MICRON CHMOS 12-MACROCELL EPLD

- High-Performance LSI Semi-Custom Logic Alternative for Low-End Gate Arrays, TTL, and 74HC- or 74HCT SSI and MSI Logic, and PLDs
- High Speed t_{PD} 25 ns, 66 MHz Performance Pipelined, 33.3 MHz w/Feedback
- 12 Macrocells with Programmable I/O Architecture; Up To 22 Inputs (10 Dedicated, 12 I/O)
- 8 Programmable Inputs Configurable as Latches, Registers, or Flow-Through
- Flow-Through Input or Global CLK Pin; 1 Flow-Through Input or Global ILE/ICLK Pin
- Programmable AND, Allocatable OR Design Allows up to 16 P-Terms per Macrocell
- Software-Supported P-Term Allocation Between Adjacent Macrocells
- Programmable Output Registers Configurable as D, T, JK, or SR Types
- Dual Feedback on All Macrocells for Implementing Buried Registers with Bidirectional I/O
- 2 P-Terms on All Macrocell Control Signals
- Programmable Low-Power Option for Standby Operation; 100 μA Typical Standby Current
- UV Erasable (CerDIP) EPROM Technology or OTP
- 100% Generically Tested EPROM Logic Control Array
- Programmable Security Bit Allows 100% Protection of Proprietary Designs
- Available in 24-Pin 300-mil CerDIP/PDIP and 28-Pin PLCC Packages

(See Packaging Spec., Order Number #231369)

2

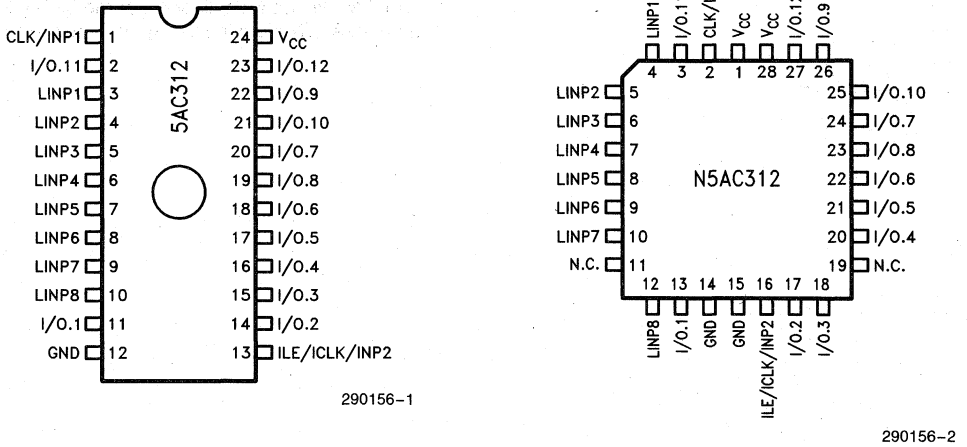


Figure 1. Pin Configurations

October 1990
Order Number: 290156-004

INTRODUCTION

The Intel 5AC312 CHMOS EPLD (Erasable Programmable Logic Device) represents an innovative approach to overcoming the primary limitations of standard PLDs. Due to a proprietary I/O architecture and macrocell structure, the 5AC312 is capable of implementing high performance logic functions more effectively than previously possible. It can be used as an alternative to low-end gate arrays, multiple programmable logic devices or LS-, HC- or HCT SSI and MSI logic devices. Input and macrocell features for the 5AC312 are a superset of features offered by other PLD-type products.

The 5AC312 uses advanced CHMOS EPROM cells as logic control elements instead of poly-silicon fuses. This technology allows the 5AC312 to operate at levels necessary in high performance systems while significantly reducing the power consumption. Its programmable stand-by function reduces power consumption to almost "zero" in applications where a slight speed loss is traded for power savings.

ARCHITECTURE DESCRIPTION

The architecture of the 5AC312 is based on the familiar "Sum-Of-Products" programmable AND, fixed OR structure, though the 5AC312 macrocell contains a number of significant functional enhancements. This device can implement both combinational and sequential logic functions through a highly flexible macrocell and I/O structure. The 5AC312 has been designed to effectively implement both combinational-register and register-combinational-register forms of logic to easily accommodate state machine designs.

Figure 2 shows a global view of the 5AC312 architecture. The 5AC312 contains a total of 12 I/O macrocells, 8 user-programmable input structures, and 2 additional inputs that can be programmed to serve as either combinatorial inputs or clock inputs. Each of the eight inputs can be individually configured as a latch, register, or flow-through input. Input latches/registers can be synchronously or asynchronously clocked.

Each macrocell is further sub-divided into 16 Product Terms with 8 Product Terms dedicated to the control signals OE, PRESET, ASYNCH. CLK and CLEAR, and 8 Product Terms available for the general data array (see Figure 3).

The basic macrocell architecture of the 5AC312 includes a user-programmable AND array and a user-configurable OR array. The inputs to the programmable AND array originate from the true and complement signals from the programmable input structure, the dedicated inputs, and the 24 feedback paths from the 12 I/O macrocells.

PROGRAMMABLE INPUTS

Figure 4 shows a block diagram of the 5AC312 input architecture. This device contains 8 user-program-

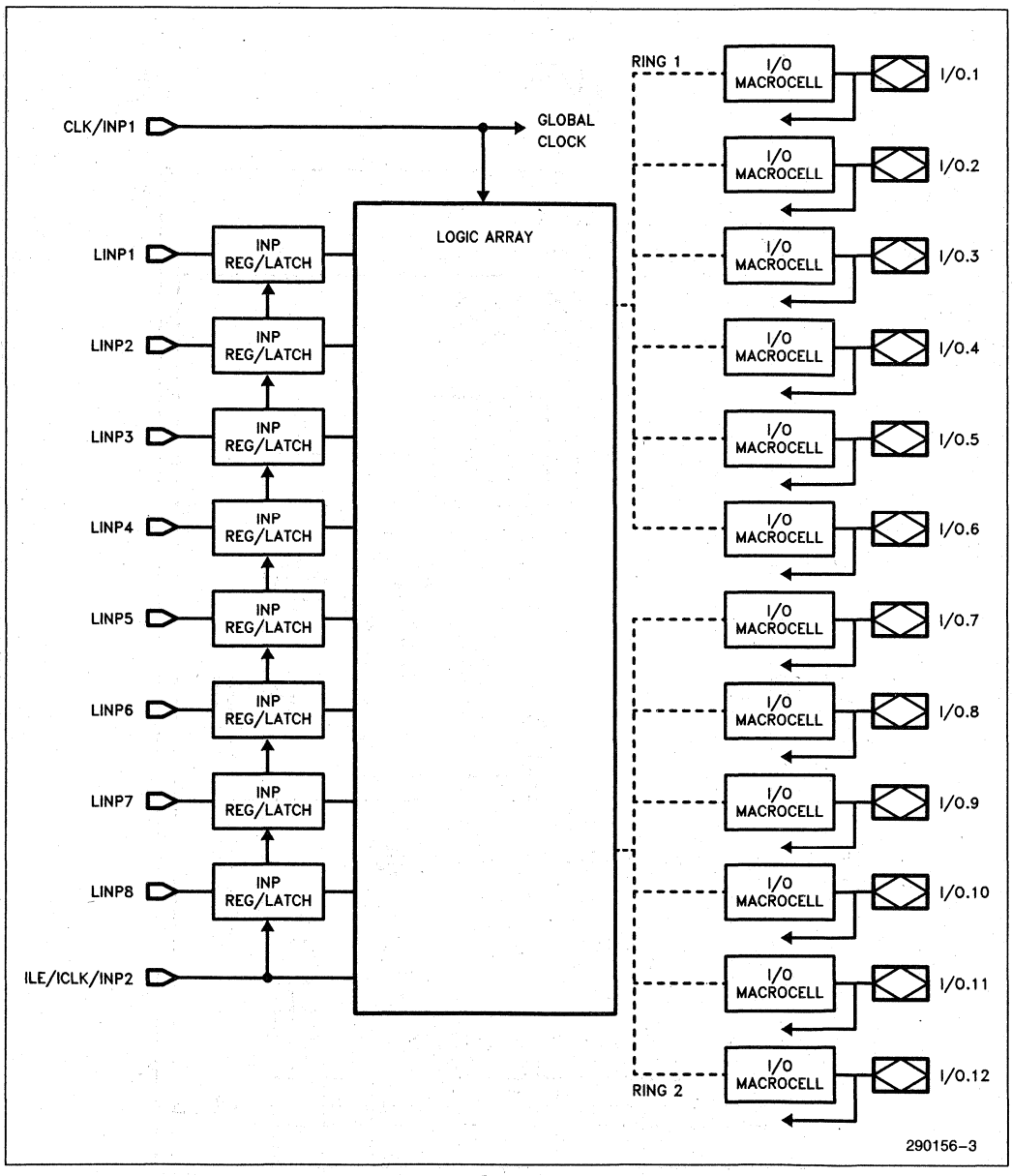


Figure 2. 5AC312 Architecture

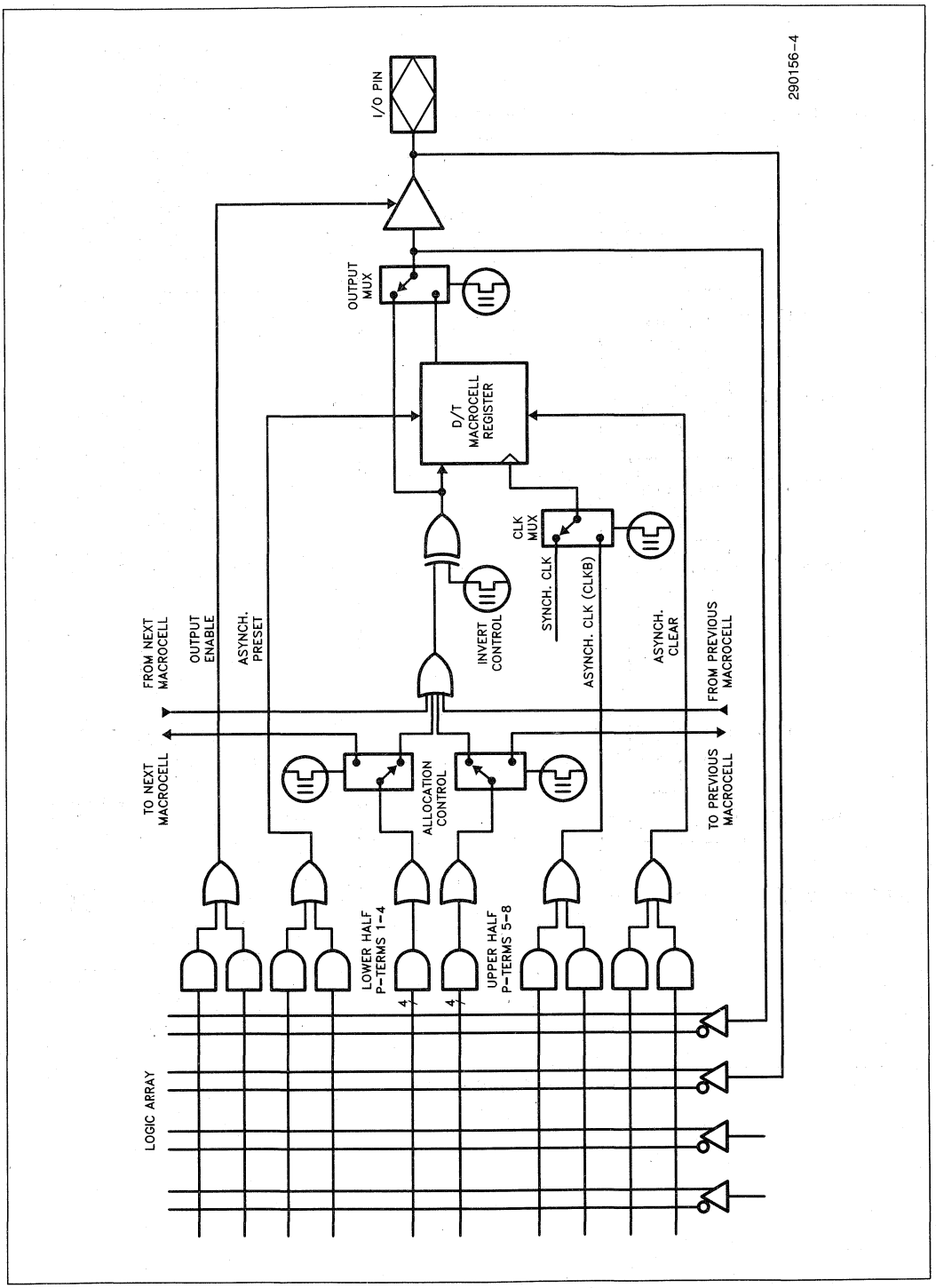
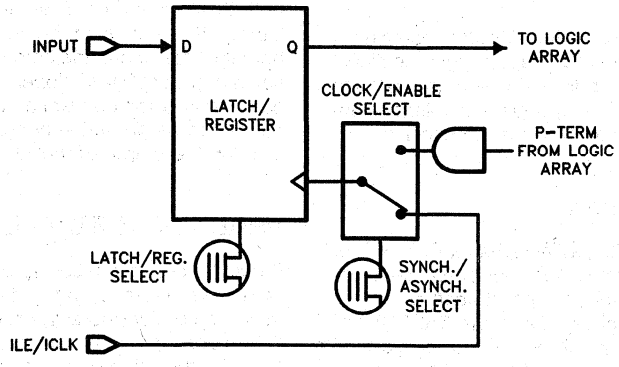


Figure 3. 5AC312 Basic Macrocell Structure



NOTE:
Software implements a direct (flow-through) input by selecting an asynchronous latch and tying its control P-term to V_{CC}.

Figure 4. 5AC312 Input Structure

mable input structures that can be individually configured to work in one of five modes:

- Input register (D-register), synchronous operation
- Input register (D-register), asynchronous operation
- Input latch (D-latch), synchronous operation
- Input latch (D-latch), asynchronous operation
- Flow-through input

The configuration is accomplished through the programming of EPROM architecture control bits by the logic compiler and programmer software. If synchronous operation is chosen, the ILE/ICLK/INP becomes an ILE/ICLK (Input Latch Enable) input global to all input latch/register structures. For asynchronous operation, ILE/ICLK/INP can be used as a normal input (flow-through input) to the device while a separate Product Term in the control array is used to derive an input clock signal for the input structure. Because the clock signal for each input structure can be individually selected, a mix between synchronously and asynchronously clocked input structures is also possible.

MACROCELLS

Each of 12 macrocells in the 5AC312 contains 8 p-terms (Product Terms) to support logic functions. These 8 p-terms are subdivided into 2 groups each containing 4 p-terms. This grouping of p-terms supports the proprietary p-term allocation scheme.

Register Configuration

Each macrocell can be configured as a D, T, RS, or JK register. The 8 p-terms for control functions are organized so that 2 p-terms support *each of the four* control signals. Control signals in the 5AC312 are: Output Enable (OE), asynchronous I/O register preset (PRESET), asynchronous clock for I/O registers (ASYNCH. CLK), and asynchronous I/O register reset (CLEAR). Availability of 2 p-terms per control signal is another feature that increases the efficiency of the device by reducing the need to use intermediate macrocells sometimes needed to implement control functions.

CLK is a global clock signal that can be used to synchronously clock any or all macrocell registers. It can be used as an input to the logic array at the same time as a macrocell clock. When CLK is not used as a synchronous clock, it functions only as a dedicated input to the logic array.

Combinatorial Configuration

The macrocell register can be bypassed to implement combinatorial logic functions. When configured to provide combinatorial logic, only the OE control signal is used.

Invert Select Bit

An invert select EPROM bit is used to invert the product term input into each macrocell register, including double inputs on JK and SR registers. This invert option allows the highest possible logic utilization by use of DeMorgan's logic inversion.

LOGIC ARRAY

Each intersecting point in the logic array contains a programmable EPROM connection. Initially (erased state), all connections are complete, i.e., both true and complement states of all signals are connected to each p-term. Connections are opened during programming. When both the true and complement connections exist, a logical false results on the output of the AND gate. If both the true and complement connections of a signal are programmed "open", then a logic "don't care" results for that signal. If all connections for a p-term are programmed open, then a logical true results on the output of the AND gate.

PRODUCT TERM ALLOCATION

Product Term allocation is defined as taking logic resources (p-terms) away from macrocells where they are not used to support demand for more than 8 Product Terms in other areas of the chip. In the 5AC312, this allocation can occur in increments of 4 p-terms between adjacent macrocells.

The 12 macrocells available in the 5AC312 are grouped into two "rings" with 6 macrocells per ring. Product Terms can be allocated in a "shift register" mode inside a ring; allocation of Product Terms between the rings is not supported. The two rings are shown in Figure 2 and listed in Table 1.

Example:

The logic function in macrocell 4 requires 16 p-terms. In this case, the iPLS II software allocates 4 p-terms from the previous macrocell in Ring 1 (macrocell 3) and 4 p-terms from the next macrocell in Ring 1 (macrocell 5) to accumulate a total of 16 p-terms (8 + 4 + 4). This implementation leaves macrocells 3 and 5 with a remainder of 4 p-terms each. These remaining p-terms in macrocells 3 and 5 can also be allocated away to or can be supplemented with p-terms from their respective previous/next macrocells in Ring 1.

Applying this scheme to the 5AC312 it becomes clear that any macrocell inside the device can support logic functions requiring between 0 and 16 Product Terms. Product Terms allocated away from a macrocell do not affect that macrocell's output structure. If all Product Terms are allocated "away" from a macrocell, the input to that macrocell's I/O control block is tied to GND. This polarity can be changed by programming the invert select EPROM bit. The I/O register as well as all secondary controls to this I/O control block are still available and can be used if needed.

The Product Term allocation scheme described above is automatically supported by iPLS II V2.0 and is transparent to the user. Users can still use explicit pin assignments, but should assign pins in a way that does not conflict with p-term allocation. Software support allows the control signals on macrocells to be used to implement simple logic functions even when all the input p-terms have been allocated to adjacent macrocells.

Table 1. Product Term Allocation Rings

Ring 1			Ring 2		
Current Macrocell	Next Macrocell	Previous Macrocell	Current Macrocell	Next Macrocell	Previous Macrocell
1	2	6	7	8	12
2	3	1	8	9	7
3	4	2	9	10	8
4	5	3	10	11	9
5	6	4	11	12	10
6	1	5	12	7	11

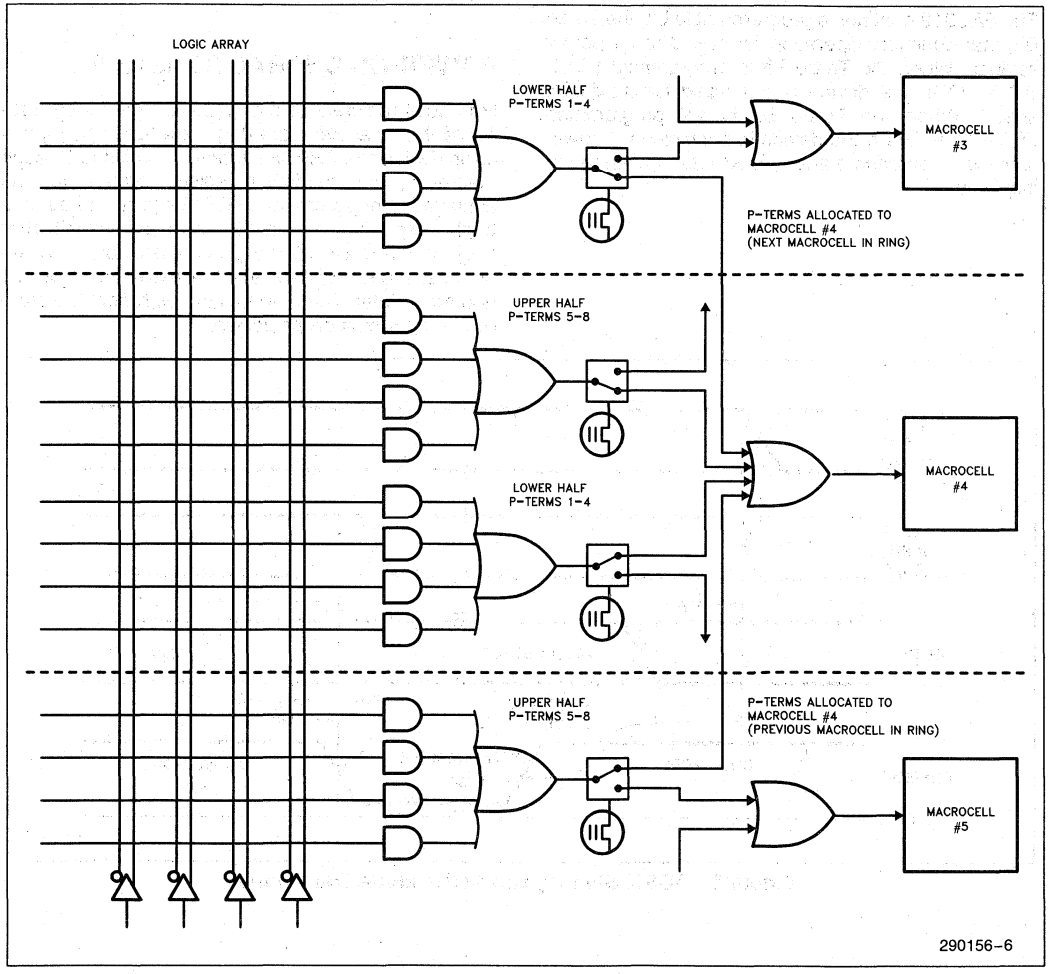


Figure 5. Product Term Allocation (8 + 4 + 4)

DUAL-FEEDBACK/BURIED LOGIC

Macrocell output can be fed back to the logic array on either one of the two feedback paths. If the pin feedback is used (connected after the output buffer), bidirectional I/O can be implemented. If the internal feedback path is used to implement a buried register or buried logic function, the pin feedback is still available for use as an input. The availability of dual feedbacks on the 5AC312 enhances resource efficiency over single feedback devices.

AUTOMATIC STANDBY MODE

The 5AC312 contains a programmable bit, the Turbo Bit, that optimizes operation for speed or for power savings. When the Turbo Bit is programmed (TURBO = ON), the device is optimized for maximum speed. When the Turbo Bit is not programmed (TURBO = OFF), the device is optimized for power savings by entering standby mode during periods of inactivity.

Figure 6 shows the device entering standby mode approximately 100 ns after the last input or I/O transition. When the next input or I/O transition is detected, the device returns to active mode. Wakeup time adds an additional 20 ns to the propagation delay through the device as measured from the first transition. No delay will occur if an output is dependent on more than one input and the last of the inputs changes after the device has returned to active mode.

After erasure, the Turbo Bit is unprogrammed (OFF); automatic standby mode is enabled. When the Turbo Bit is programmed (ON), the device never enters standby mode.

POWER-ON CHARACTERISTICS

Macrocell registers of the 5AC312 experience a reset to their inactive state (logic low) upon V_{CC} power-up. Using the PRESET function available to each macrocell, any particular register preset can be achieved after power-up. 5AC312 inputs and outputs begin responding within 10 μs (6 μs typical) after V_{CC} power-up or after a power-loss/power-up sequence. Input registers are not reset on power-up and are indeterminate. Input latches reflect the state of the input pins on power-up.

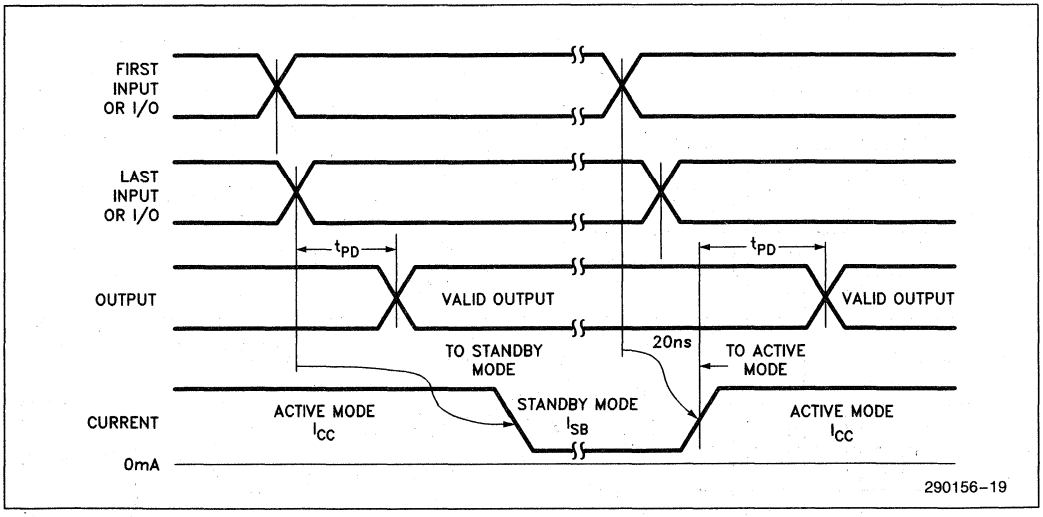


Figure 6. 5AC312 Standby and Active Mode Transitions

ERASED STATE CONFIGURATION

After erasure and prior to programming, all macrocells are configured as combinatorial, inverted outputs with output buffers three-stated. Inputs are configured as synchronous registers.

ERASURE CHARACTERISTICS

Erasure time for the 5AC312 is 1 hour at 12,000 $\mu W/cm^2$ with a 2537Å UV lamp.

Erasure characteristics of the device are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000Å. It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000Å–4000Å range. Data shows that constant exposure to room level fluorescent lighting could erase the typical 5AC312 in approximately six years, while it would take approximately two weeks to erase the device when exposed to direct sunlight. If the device is to be exposed to these lighting conditions for extended periods of time, conductive opaque labels should be placed over the device window to prevent unintentional erasure.

The recommended erasure procedure for the 5AC312 is exposure to shortwave ultraviolet light with a wavelength of 2537Å. The integrated dose (i.e., UV intensity $\times$ exposure time) for erasure should be a minimum of forty (40) $Wsec/cm^2$. The erasure time with this dosage is approximately 1 hour using an ultraviolet lamp with a 12,000 $\mu W/cm^2$ power rating. The device should be placed within 1 inch of the lamp tubes during exposure. The maximum integrated dose the 5AC312 can be exposed to without damage is 7258 $Wsec/cm^2$ (1 week at 12,000 $\mu W/cm^2$). Exposure to high intensity UV light for longer periods may cause permanent damage to the device.

int$_e$ligent Programming™ Algorithm

The 5AC312 supports the int$_e$ligent Programming algorithm which rapidly programs Intel PLDs, while maintaining a high degree of reliability. This method ensures reliability as the incremental program margin of each bit has been verified in the programming process. (Programming information for the 5AC312 is available from Intel by request.)

DESIGN SECURITY

A Security Bit provides a programmable security option to protect the data programmed in the device. Once this bit is set during programming, subsequent attempts to read the device architecture information are prevented. This method provides a higher degree of design security than fuse-based devices, since programmed EPROM cells are invisible even to microscopic examination. The Security Bit (also called the Verify Protect Bit), along with all the other EPROM cells, is reset by erasing the device.

LATCH-UP IMMUNITY

All of the input, I/O, and clock pins of the device have been designed to resist latch-up which is inherent in inferior CMOS structures. The 5AC312 is designed with Intel's proprietary 1-micron CHMOS EPROM process. Thus, each of the pins will not experience latch-up with currents up to ± 100 mA and voltages ranging from $-0.5V$ to $(V_{CC} + 0.5V)$. The programming pin is designed to resist latch-up to the 13.5 maximum device limit.

DESIGN RECOMMENDATIONS

For proper operation, it is recommended that all input and output pins be constrained to the voltage range $(GND < (V_{IN} \text{ or } V_{OUT}) < V_{CC})$. All unused inputs and I/Os should be tied to V_{CC} or GND to minimize power consumption (do not leave them floating). A power supply decoupling capacitor of at least 0.2 μF must be connected directly between each V_{CC} and GND pin.

As with all CMOS devices, ESD handling procedures should be used with the 5AC312 to prevent damage to the device during programming, assembly, and test.

FUNCTIONAL TESTING

Since the logical operation of the 5AC312 is controlled by EPROM elements, the device is completely testable during the manufacturing process. Each programmable EPROM bit controlling the internal logic is tested using application-independent test patterns. EPROM cells in the 5AC312 are 100% tested for programming and erase. After testing, the devices are erased before shipments to the customers. No post-programming tests of the EPROM array are required.

The testability and reliability of EPROM-based programmable logic devices are important features over similar devices based on fuse technology. Fuse-based programmable logic devices require a user to perform post-programming tests to insure device functionality. During the manufacturing process, tests on these parts can only be performed in very restricted manners to prevent pre-programming of the array.

INTEL PROGRAMMABLE LOGIC DEVELOPMENT SYSTEM II (iPLDS II)

Release 2.0 (or later) of iPLDS II provides all the tools needed to design with the 5AC312 EPLD. In addition to providing development assistance, iPLDS II insulates the user from knowing the intricate details of EPLD architecture (the software will optimize a design to benefit from architectural features). It minimizes logic, does automatic pin assignments and produces the best design fit for the selected EPLD. It is user friendly with guided menus, on-line Help messages and soft key inputs.

In addition, iPLDS II contains programmer hardware in the form of an iUP-PC Universal Programmer-Personal Computer to enable the user to program EPLDs, read and verify programmed devices and also to graphically edit programming files. The software generates industry standard JEDEC object code output files which can be downloaded to other programmers as well.

iPLS II software interfaces to several schematic capture packages to enable designs to be entered in schematic form. IPLDview-286/IPLDdraw allows the designer to use familiar TTL symbols or EPLD design primitive symbols. User-defined symbols are also supported. IPLDdraw also provides a path to A.C. timing simulation of EPLD designs.

SCHEMA III-PLD allows the designer to use TTL symbols, EPLD custom macros, or EPLD design primitive symbols. It also supports user-defined symbols.

Other design formats include Boolean equation entry (supported directly by iPLS II) and state machine entry (supported by iSTATE).

Detailed information on the Intel Programmable Logic Development System II is contained in a separate Intel data sheet. (Order Number: 280168). Refer to the tools section of the *Programmable Logic* handbook for a complete listing of development tools.

The 5AC312 is also supported by third-party logic compilers such as ABEL*, CUPL*, PLDesigner*, Log/IC, etc. Programming support is provided by third-party programmer companies such as Data I/O, Logical Devices, STAG, etc. Please refer to the "Third-Party Support" lists in the *Programmable Logic* handbook for complete information and vendor contacts.

ADF PRIMITIVES SUPPORTED

The following ADF primitives are supported by this device:

INP	NOTF
LINP	JOJF
RINP	JONF
CONF	SONF
COCF	SOSF
COIF	TOIF
RONF	TONF
ROIF	TOTF
RORF	CLKB
NOCF	LINB
NORF	
NOJF	
NOSF	

ORDERING INFORMATION

t_{PD} (ns)	t_{CO} (ns)	f_{MAX} (MHz)	Order Code	Package	Operating Range
25	15	66	D5AC312-25	†CERDIP	Commercial
			P5AC312-25	PDIP	
			N5AC312-25	PLCC	
30	18	50	D5AC312-30	†CERDIP	Commercial
			P5AC312-30	PDIP	
			N5AC312-30	PLCC	

†Windowed package allows UV erase.

*ABEL is a trademark of Data I/O, Corporation. CUPL is a trademark of Logical Devices, Inc. PLDesigner is a trademark of MINC, Inc.

ABSOLUTE MAXIMUM RATINGS*

Supply Voltage (V_{CC}) (1) . . . . . −2.0V to +7.0V

Programming Supply Voltage (V_{PP}) (1) . . . . . −2.0V to +13.5V

D.C. Input Voltage (V_I)(1, 2) . . . −0.5V to V_{CC} + 0.5V

Storage Temperature (T_{stg}) . . . . . −65°C to +150°C

Ambient Temperature (T_{amb}) (3) . . −10°C to +85°C

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

NOTES:
1. Voltages with respect to GND.
2. Minimum D.C. input is −0.5V. During transitions, the inputs may undershoot to −2.0V or overshoot to +7V for periods of less than 20 ns under no load conditions.
3. Under bias. Extended temperature range versions are available.

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Min	Max	Unit
V_{CC}	Supply Voltage	4.75	5.25	V
V_{IN}	Input Voltage	0	V_{CC}	V
V_O	Output Voltage	0	V_{CC}	V
T_A	Operating Temperature	0	+70	°C
t_R	Input Rise Time		500	ns
t_F	Input Fall Time		500	ns

D.C. CHARACTERISTICS T_A = 0°C to +70°C, V_{CC} = 5.0V ±5%

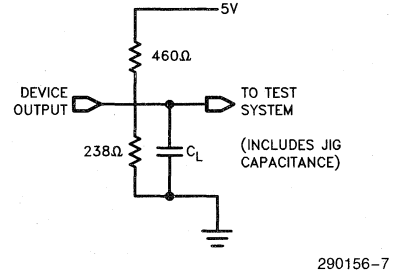
Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
V_{IH}(4)	High Level Input Voltage	2.0		V_{CC} + 0.3	V	
V_{IL}(4)	Low Level Input Voltage	−0.3		0.8	V	
V_{OH}(5)	High Level Output Voltage	2.4			V	I_O = −4.0 mA D.C., V_{CC} = min.
V_{OL}	Low Level Output Voltage			0.45	V	I_O = 8.0 mA D.C., V_{CC} = min.
I_I	Input Leakage Current			±10	μA	V_{CC} = max., GND < V_{IN} < V_{CC}
I_{OZ}	Output Leakage Current			±10	μA	V_{CC} = max., GND < V_{OUT} < V_{CC}
I_{SC}(6)	Output Short Circuit Current	−30		−90	mA	V_{CC} = max., V_{OUT} = 0.5V
I_{SB}(7)	Standby Current		100	300	μA	V_{CC} = max., V_{IN} = V_{CC} or GND, Standby Mode
I_{CC}	Power Supply Current (See I_{CC} vs Freq. Graph)		10		mA	V_{CC} = max., V_{IN} = V_{CC} or GND, No Load, Input Freq. = 1 MHz Active Mode (Turbo = Off), Device Prog. as 12-Bit Ctr.

NOTES:
4. Absolute values with respect to device GND; all over and undershoots due to system or tester noise are included. Do not attempt to test these values without suitable equipment.
5. I_O at CMOS levels (3.84V) = −2 mA.
6. Not more than 1 output should be tested at a time. Duration of that test must not exceed 1 second.
7. With Turbo Bit Off, device automatically enters standby mode approximately 100 ns after last input transition.

CAPACITANCE

Symbol	Parameter	Min	Typ	Max	Unit	Conditions
C_{IN}	Input Capacitance			8	pF	V_{IN} = 0V, f = 1.0 MHz
C_{OUT}	I/O Capacitance			15	pF	V_{OUT} = 0V, f = 1.0 MHz
C_{CLK}	ILE/ICLK/INP2 Capacitance			12	pF	V_{IN} = 0V, f = 1.0 MHz
C_{VPP}	V_{PP} Pin (CLK/INP1)			25	pF	V_{IN} = 0V, f = 1.0 MHz

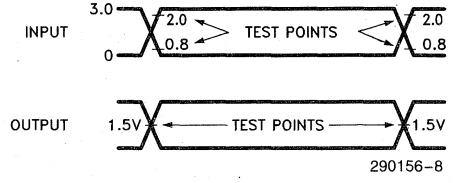
A.C. TESTING LOAD CIRCUIT



290156-7

C_L = 30 pF

A.C. TESTING INPUT, OUTPUT WAVEFORM



290156-8

A.C. Testing: Inputs are driven at 3.0V for a Logic "1" and 0V for a Logic "0". Timing Measurements are made at 2.0V for a Logic "1" and 0.8V for a Logic "0" on inputs. Outputs are measured at a 1.5V point. Device input rise and fall times < 6 ns.

A.C. CHARACTERISTICS T_A = 0°C to +70°C, V_{CC} = 5.0V ±5%, Turbo Bit "On"(8)

Symbol	From	To	5AC312-25			5AC312-30			Non-(10) Turbo Mode	Unit
			Min	Typ	Max	Min	Typ	Max		
t_{PD}	Input or I/O	Comb. Output		20	25		25	30	+20	ns
t_{PZX}(9)	Input or I/O	Output Enable		20	25		25	30	+20	ns
t_{PXZ}(9)	Input or I/O	Output Disable		20	25		25	30	+20	ns
t_{CLR}	Asynch. Reset	Q Reset		20	25		25	30	+20	ns
t_{SET}	Asynch. Set	Q Set		20	25		25	30	+20	ns

NOTES:
8. Typical values are at T_A = 25°C, V_{CC} = 5V, Active Mode.
9. t_{PZX} and t_{PXZ} are measured at ±0.5V from steady-state voltage as driven by spec. output load. t_{PXZ} is measured with C_L = 5 pF.
10. If device is operated with Turbo Bit Off (Non-Turbo Mode) and the device has been inactive for approximately 100 ns, increase time by amount shown.

SYNCHRONOUS CLOCK MODE (MACROCELLS) A.C. CHARACTERISTICS

T_A = 0°C to +70°C, V_{CC} = 5.0V ±5%, Turbo Bit On[8]

Symbol	Parameter	5AC312-25			5AC312-30			Non-[10] Turbo Mode	Unit
		Min	Typ	Max	Min	Typ	Max		
f_{MAX}	Max. Frequency (Pipelined) $1/t_{SU}$—No Feedback		80	66		66	50		MHz
f_{CNT1}	Max. Count Frequency $1/(t_{SU} + t_{CO})$—External Feedback		40	33.3		33	26.3		MHz
f_{CNT2}	Max. Count Frequency $1/t_{CNT}$—Internal Feedback		40	33.3		35	28.5		MHz
t_{SU1}	Input Setup Time to CLK ↑	15	12		20	18		+20	ns
t_{SU2}	I/O Setup Time to CLK ↑	15	12		20	18		+20	ns
t_H	I or I/O Hold after CLK ↑	0			0				ns
t_{CO}	CLK ↑ to Output Valid		10	15		12	18		ns
t_{CNT}	Macrocell Output Feedback to Macrocell Input—Internal Path		25	30		30	35	+20	ns
t_{CH}	CLK High Time	7			9				ns
t_{CL}	CLK Low Time	7			9				ns
t_{CW}	Minimum Clock Period	15			20				ns

SYNCHRONOUS CLOCK MODE (INPUT STRUCTURE) A.C. CHARACTERISTICS

T_A = 0°C to +70°C, V_{CC} = 5.0V ±5%, Turbo Bit On[8]

Symbol	Parameter	5AC312-25			5AC312-30			Non-[10] Turbo Mode	Unit
		Min	Typ	Max	Min	Typ	Max		
f_{MAXI}	Max. Frequency ($1/t_{CWI}$)		80	66		66	50		MHz
t_{SUIR}	Input Register/Latch Setup Time before ILE/ICLK ↓	5			5				ns
t_{ESUI}[11]	Input Latch Setup Time before ILE ↑	5			5				ns
t_{COI}	ICLK ↓ to Comb. Output		30	35		35	40	+20	ns
t_{EOI}	ILE ↑ to Comb. Output		30	35		35	40	+20	ns
t_{HI}	Input Hold after ICLK/ILE ↓	7			10				ns
t_{EHI}	Input Hold after ILE ↓	7			10				ns
t_{CHI}	ILE/ICLK High Time	7			9				ns
t_{CLI}	ILE/ICLK Low Time	7			9				ns
t_{CWI}	Minimum Input Clock Period	15			20				ns

NOTE:
11. This specification must be met to guarantee t_{EOI}. When ILE goes high before data is valid, use t_{PD} instead of t_{EOI}.

ASYNCHRONOUS CLOCK MODE INPUT STRUCTURE A.C. CHARACTERISTICS

T_A = 0°C to +70°C, V_{CC} = 5.0V ±5%, Turbo Bit On[8]

Symbol	Parameter	5AC312-25			5AC312-30			Non-[10] Turbo Mode	Unit
		Min	Typ	Max	Min	Typ	Max		
f_{AMAXI}	Max. Frequency Input Register $1/(t_{ACLI} + t_{ACHI})$		80	66		66	50		MHz
t_{ASUIR}	Input Register/Latch Setup Time to Asynch. ILE/ICLK	0			0				ns
t_{AESUI}[11]	Input Latch Setup Time before Asynch. ILE	0			0				ns
t_{ACOI}	Asynch. ICLK to Comb. Output		40	48		45	55	+20	ns
t_{AEOI}	Asynch. ILE ↑ to Comb. Output		40	48		45	55	+20	ns
t_{AHI}	Input Register/Latch Hold after Asynch. ILE/ICLK	20	14		25	20		+20	ns
t_{AEHI}	Input Hold after Asynch. ILE	20	14		25	20			ns
t_{ACHI}	Asynch. ICLK High Time	7			9			+20	ns
t_{ACLI}	Asynch. ICLK Low Time	7			9			+20	ns
t_{ACWI}	Minimum Input Clock Period	15			20			+20	ns

ASYNCHRONOUS CLOCK MODE MACROCELLS A.C. CHARACTERISTICS

T_A = 0°C to +70°C, V_{CC} = 5.0V ±5%, Turbo Bit On[8]

Symbol	Parameter	5AC312-25			5AC312-30			Non-[10] Turbo Mode	Unit
		Min	Typ	Max	Min	Typ	Max		
f_{AMAX}	Max. Frequency (Pipelined) $1/(t_{ACL} + t_{ACH})$—No Feedback		80	66		66	50		MHz
f_{ACNT1}	Max. Frequency $1/(t_{ASU} + t_{ACO})$—External Feedback		35	28.5		33	23.8		MHz
f_{ACNT2}	Max. Frequency $1/t_{ACNT}$—with Feedback		40	33.3		35	30		MHz
t_{ASU1}	Input Setup Time to Asynch. Clock	10			12			+20	ns
t_{ASU2}	I/O Setup Time to Asynch. Clock	10			12			+20	ns
t_{AH}	Input or I/O Hold after Asynch. Clock	5	0		5	0			ns
t_{ACO}	Asynch. CLK to Output Valid		20	25		25	30	+20	ns
t_{ACNT}	Register Output Feedback to Register Input—Internal Path		25	30		30	35	+20	ns
t_{ACH}	Asynch. CLK High Time	7			9			+20	ns
t_{ACL}	Asynch. CLK Low Time	7			9			+20	ns
t_{ACW}	Minimum Asynch. CLK Period	15			20			+20	ns

INPUT-CLOCK-TO-MACROCELL-CLOCK A.C. CHARACTERISTICS

T_A = 0°C to +70°C, V_{CC} = 5.0V ±5%, Turbo Bit On[8]

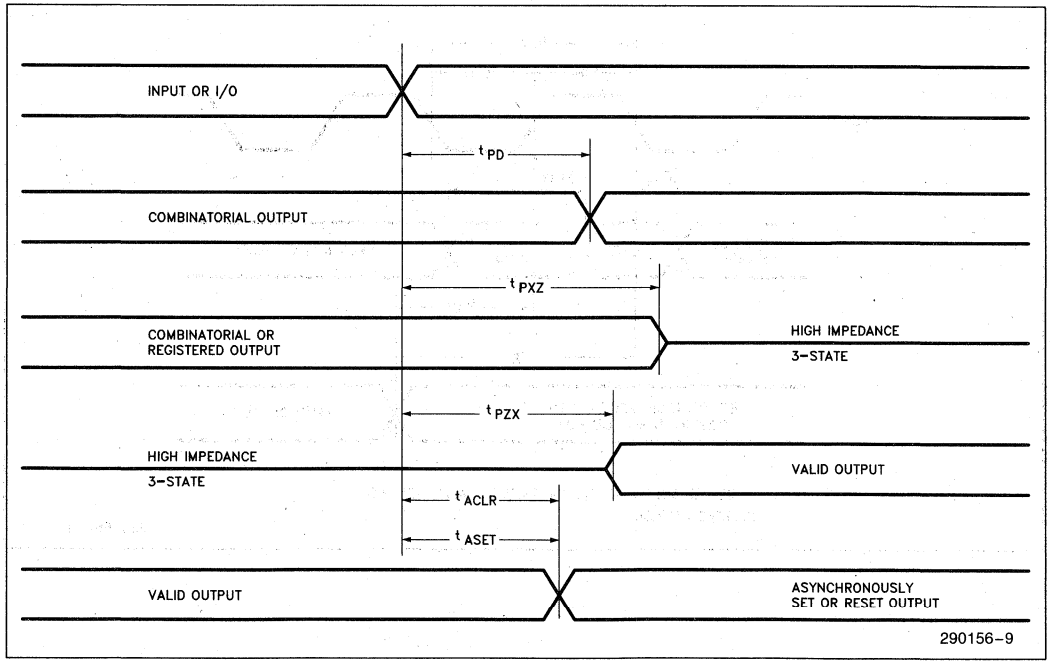
Symbol	Parameter	5AC312-25			5AC312-30			Non-[10] Turbo Mode	Unit
		Min	Typ	Max	Min	Typ	Max		
t_{C1C2}[12]	Synchronous ILE/ICLK to Synchronous Macrocell CLK	25			30			+20	ns
	Synchronous ILE/ICLK to Asynchronous Macrocell CLK	15			18			+20	ns
	Asynchronous ILE/ICLK to Synchronous Macrocell CLK	35			40			+20	ns
	Asynchronous ILE/ICLK to Asynchronous Macrocell CLK	25			35			+20	ns

NOTE:
12. Times for SETUP, HOLD, and OUTPUT VALID are shown in previous tables.

2

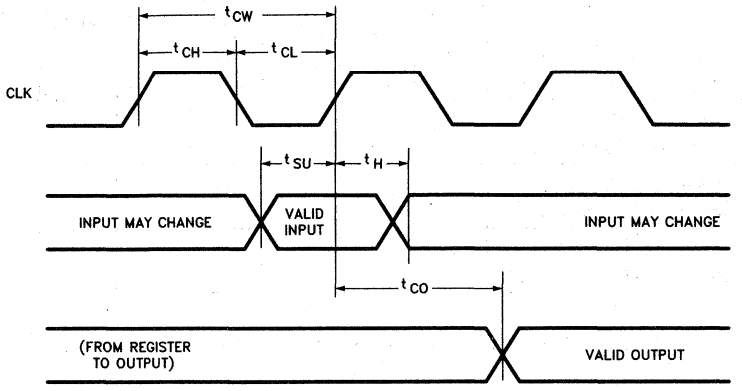
SWITCHING WAVEFORMS

COMBINATORIAL MODE



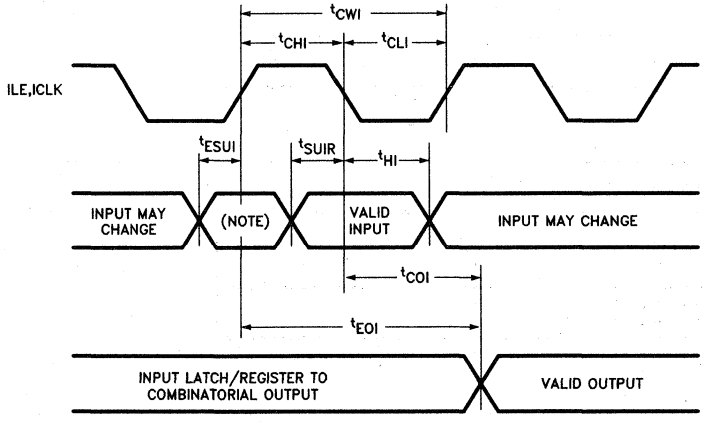
SWITCHING WAVEFORMS (Continued)

SYNCHRONOUS CLOCK MODE (MACROCELLS)



290156-10

SYNCHRONOUS CLOCK MODE (INPUT STRUCTURE)

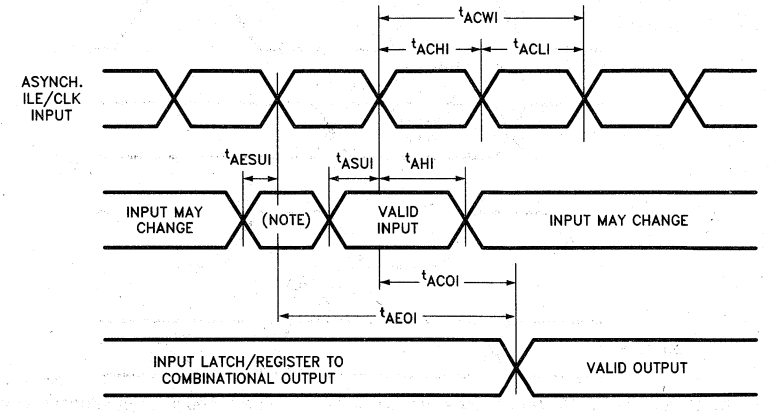


NOTE: WHEN ILE GOES HIGH BEFORE DATA IS VALID, USE t_{PD} INSTEAD OF t_{EOI}.

290156-11

SWITCHING WAVEFORMS (Continued)

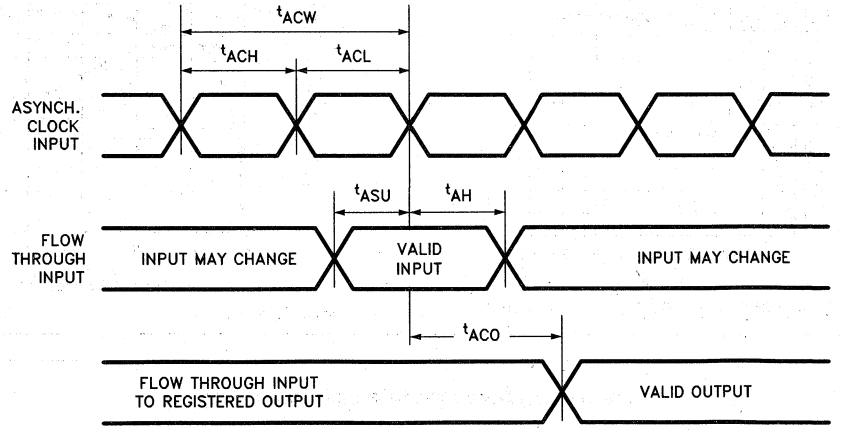
ASYNCHRONOUS CLOCK MODE (INPUT STRUCTURE)



NOTE: WHEN ILE GOES HIGH BEFORE DATA IS VALID, USE ^tPD INSTEAD OF ^tAEOI.

290156-13

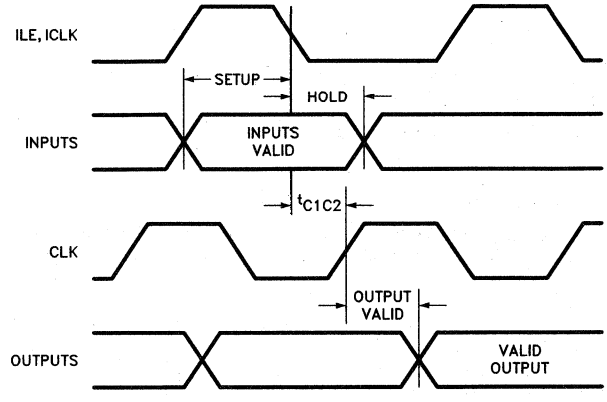
ASYNCHRONOUS CLOCK MODE (MACROCELLS)



290156-12

SWITCHING WAVEFORMS (Continued)

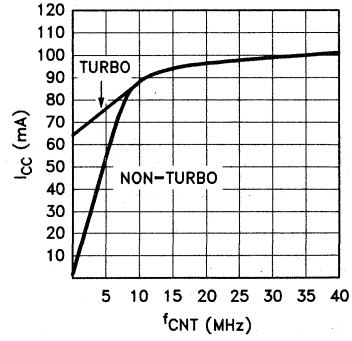
INPUT CLOCK-TO-MACROCELL CLOCK TIMING (CLOCKED PIPELINED DATA)



290156-18

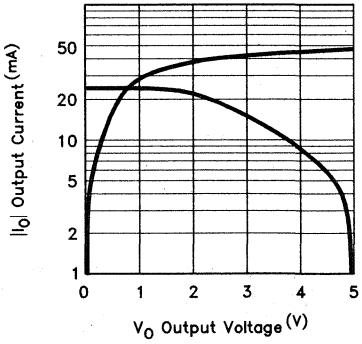
CLOCK, SETUP, HOLD, and OUTPUT VALID times are dependent on synchronous/asynchronous clocking and are listed in the specification tables.

5AC312 Current in Relation to Frequency



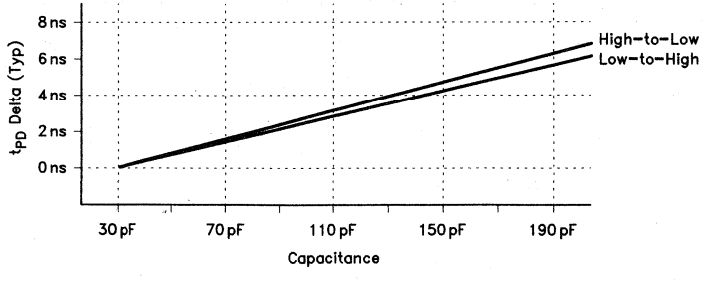
Conditions: T_A = +25°C, V_{CC} = 5.25V 290156-20

5AC312 Output Drive Current in Relation to Voltage



Conditions: T_A = +25°C 290156-16

5AC312 t_{PD} Derating vs Capacitive Loading



T_A = +25°C
V_{CC} = 5.0V

290156-21

5AC324
1-MICRON CHMOS 24-MACROCELL EPLD

- High-Performance LSI Semi-Custom Logic Alternative to Low-end Gate Arrays, TTL, 74HC SSI and MSI Logic, and PLDs
- High Speed t_{PD} 25 ns, 66 MHz Performance Pipelined, 33.3 MHz w/ Feedback
- 24 Macrocells with Programmable I/O Architecture; Up to 36 Inputs (12 Dedicated, 24 I/O)
- 10 Programmable Inputs Configurable as Latches, Registers, or Flow-Through
- 1 Flow-Through Input or Global CLK Pin; 1 Flow-Through Input or Global ILE/ICLK Pin
- Programmable AND, Allocatable OR Design Allows up to 16 P-Terms per Macrocell
- Software-Supported P-Term Allocation Between Adjacent Macrocells
- Programmable Output Registers Configurable as D, T, JK, or SR Types
- Dual Feedback on All Macrocells for Implementing Buried Registers with Bidirectional I/O
- 2 P-Terms on All Macrocell Control Signals
- Programmable Low-Power Option for "Stand-by" Operation; 150 μA Typical Standby Current
- UV Erasable (CerDIP) EPROM Technology or OTP
- 100% Generically Tested EPROM Logic Control Array
- JEDEC Pinout
- Available in 40-pin CerDIP/PDIP and 44-Pin PLCC Packages

(See Packaging Spec., Order Number #231369)

2

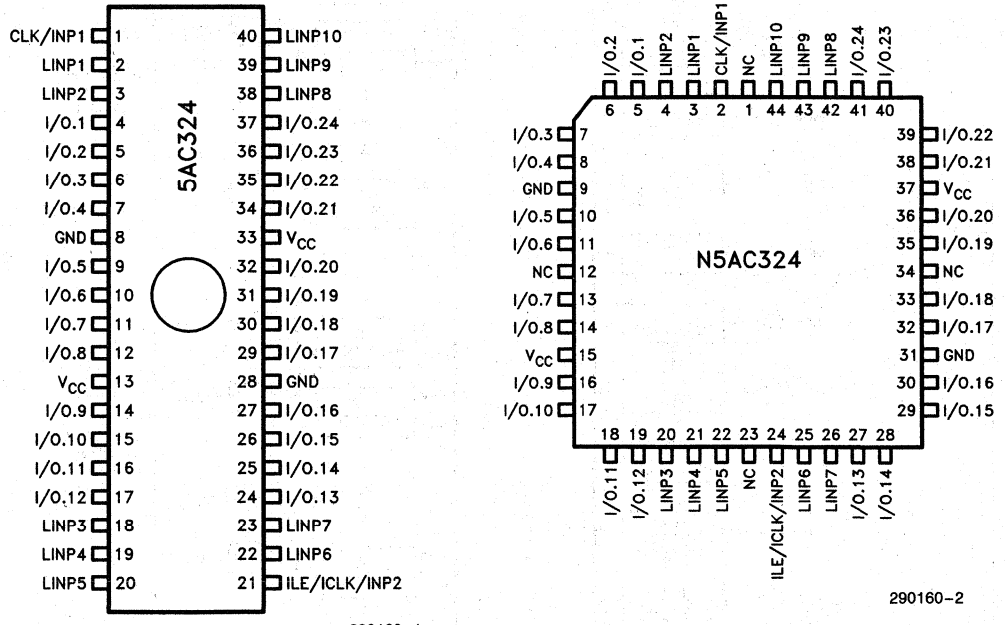


Figure 1. 5AC324 Pinout Diagrams

October 1990
Order Number: 290160-004

INTRODUCTION

The Intel 5AC324 CHMOS EPLD (Erasable Programmable Logic Device) is a high integration device that overcomes the primary limitations of standard PLDs. Due to a proprietary I/O architecture and macrocell structure, the 5AC324 is capable of implementing high performance logic functions more effectively than previously possible. The 5AC324 can be used as an alternative to low-end gate arrays, multiple programmable logic devices, or LS-, HC-, or HCT SSI and MSI logic devices. Input and macrocell features for the 5AC324 are a superset of features offered on other PLD-type products.

The 5AC324 uses advanced CHMOS EPROM cells as logic control elements instead of poly-silicon fuses. This technology allows the device to operate at levels necessary in high performance systems while significantly reducing power consumption. Its programmable standby mode reduces power to near zero in applications where a slight speed loss is traded for power savings.

ARCHITECTURE DESCRIPTION

The architecture of the 5AC324 is based on the familiar "Sum-Of-Products" programmable AND, fixed OR structure. This structure is then surrounded by powerful, programmable macrocells and inputs. The 5AC324 can implement both combinatorial and sequential logic functions through a highly flexible macrocell and I/O structure. The architecture of the device supports both combinatorial-register and register-combinatorial-register forms of logic to easily accomodate state machine designs.

Figure 2 shows a global view of the 5AC324 architecture. The 5AC324 contains a total of 24 I/O programmable macrocells, 10 programmable input structures, and two clock inputs that can be programmed to function either as combinatorial inputs or clock inputs for the input structures and macrocells.

Each of the ten programmable inputs can be individually configured as a latch, register or flow-through

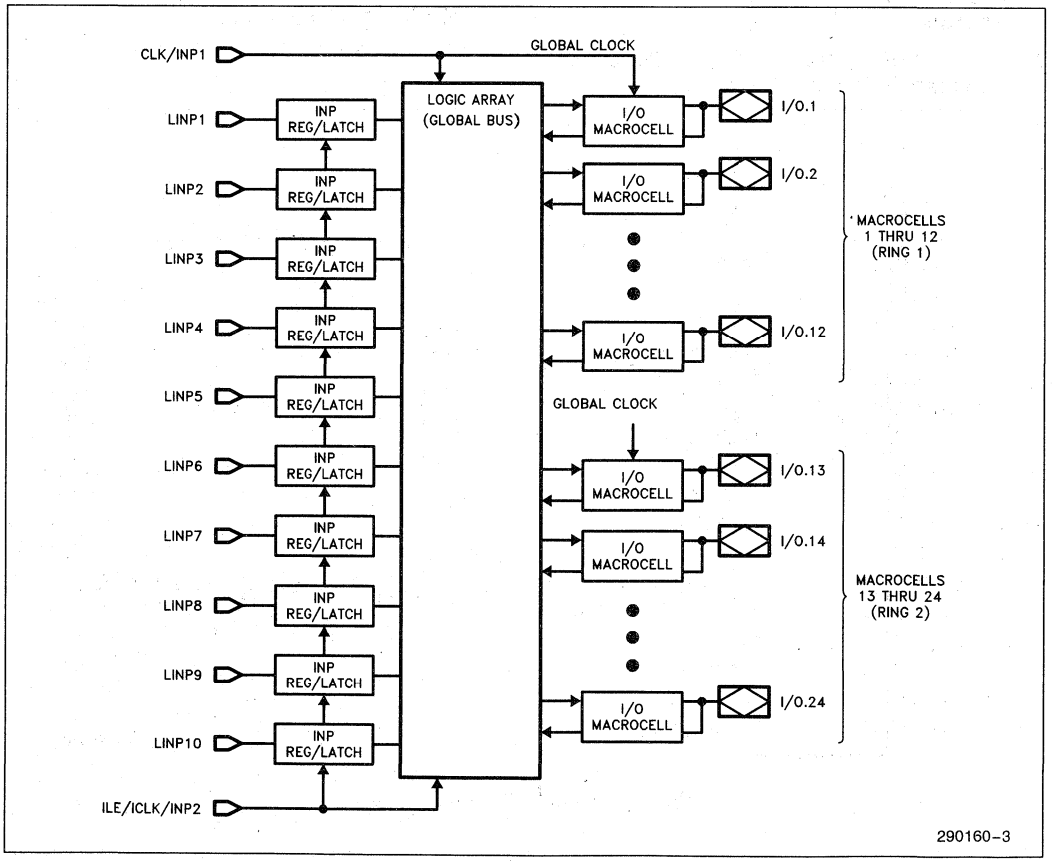


Figure 2. 5AC324 Global Architecture

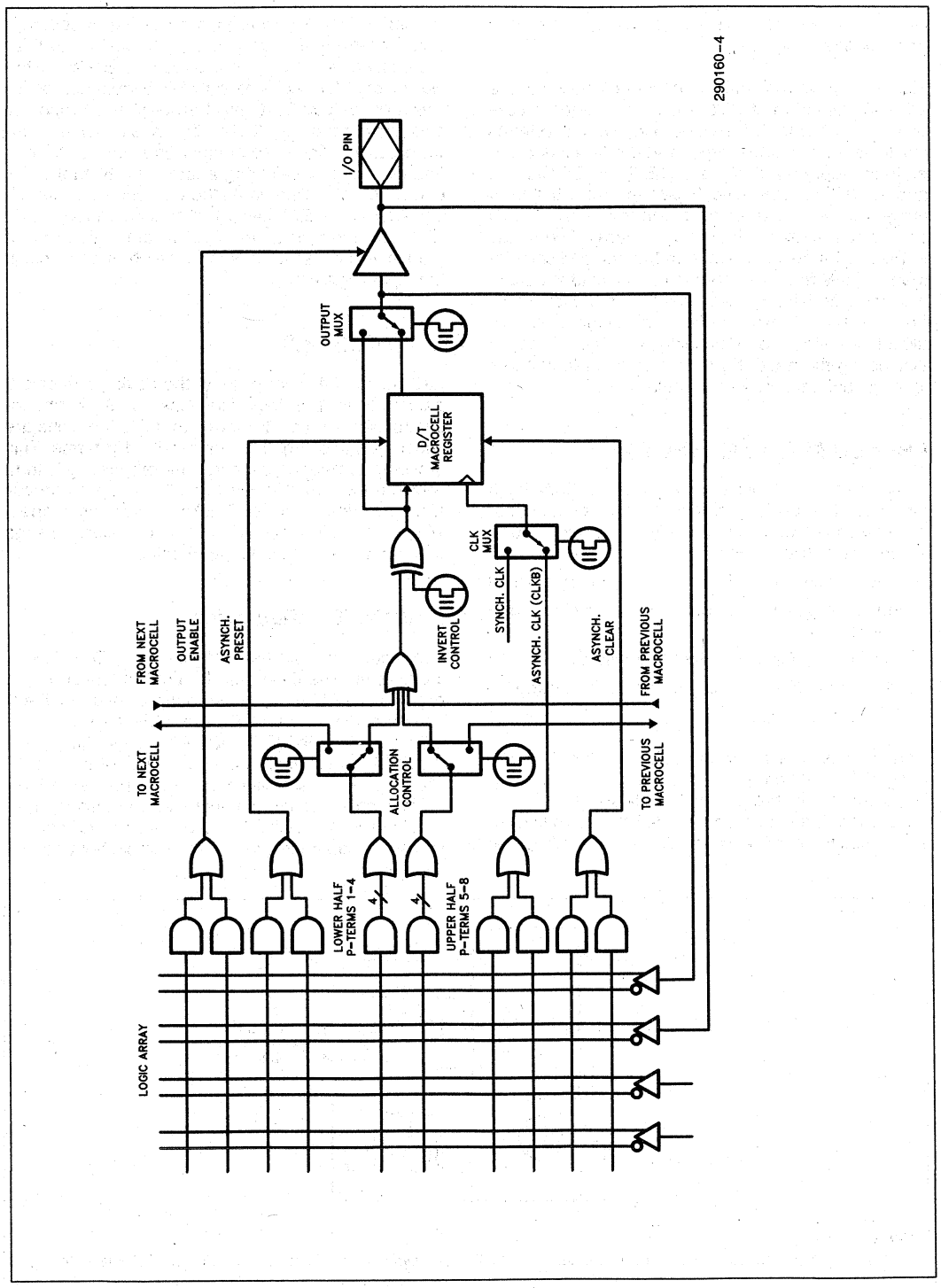


Figure 3. 5AC324 Macrocell Architecture

input. Input latches/registers can be synchronously or asynchronously clocked.

Figure 3 shows the basic architecture of each of the 24 macrocells in the 5AC324. Each macrocell contains 16 p-terms (product terms), with 8 p-terms available for the global array and 8 p-terms dedicated to the four control signals: OE, PRESET, CLEAR, and ASYNCH. CLK. The 8 p-terms from the logic array are organized as a user-programmable AND array and a user-configurable OR array. The inputs to the AND array originate from the true and complement signals from the programmable input structure, the dedicated inputs, and the 48 feedback paths from the 24 I/O macrocells to the global bus. This global bus simplifies designing with the device by eliminating the need to partition a circuit to fit into a local/global internal bus structure.

PROGRAMMABLE INPUTS

Figure 4 shows a block diagram of the 5AC324 input structure. The device contains 10 user-programmable inputs that can be individually configured to operate in one of five modes:

- input register (D-register), synchronously clocked
- input register (D-register), asynchronously clocked
- input latch, (D-latch), synchronously clocked
- input latch, (D-latch), asynchronously clocked
- Flow-through input

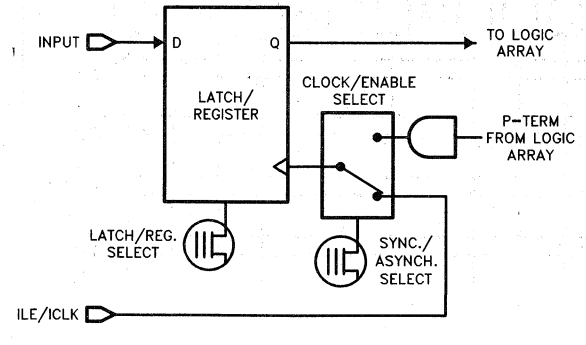
Configuration is accomplished through the programming of EPROM architecture control bits via the logic compiler and programmer software. If synchronous operation is selected, the ILE/ICLK pin is used as a global latch/clock to all input latch/register structures. For asynchronous operation, a separate product term in the array is used to derive the ILE/ICLK signal for each input structure. Because the clock signal for each programmable input can be individually selected, a mix between synchronously and asynchronously clocked inputs is possible. Software can configure each input structure as a flow-through input by selecting a latch and tying the ILE p-term to VCC. Data is latched/clocked on the falling edge of ILE/ICLK (synchronous mode). ILE/ICLK can function as an input to the logic array at the same time as it is used to synchronously clock the input registers.

MACROCELLS

Each of the 24 macrocells in the device contains 8 p-terms to support logic functions and 8 p-terms for control signals. The 8 p-terms for logic functions are subdivided into 2 groups, each with 4 p-terms. This grouping of p-terms supports the proprietary p-term allocation scheme in the 5AC324. Each macrocell also provides dual feedbacks to the logic array, which results in more efficient macrocell/pin usage than possible with single feedbacks.

Register Configuration

Each macrocell can be configured as a D, T, RS, or JK register. The 8 p-terms for control functions are organized so that 2 p-terms support each of the 4 control signals: Output Enable (OE), asynchronous I/O preset (PRESET), asynchronous I/O reset (CLEAR), and asynchronous I/O register clock (ASYNCH. CLK). Availability of 2 p-terms per control signal is another feature that increases the efficiency of the device by reducing the need to use intermediate macrocells sometimes needed to implement control functions.



NOTE:
Software implements a direct (flow-through) input by selecting an asynchronous latch and tying its control P-term to V_{CC}.

Figure 4. 5AC324 Programmable Input Structure

CLK is a global clock signal that can be used to synchronously clock any or all macrocell registers. When CLK is not used as a synchronous clock, it functions only as a dedicated input to the logic array. CLK can be used as an input to the logic array at the same time as it is used as a macrocell clock.

Combinatorial Configuration

The macrocell register can be bypassed to implement combinatorial logic functions. When configured to provide combinatorial logic, only the OE control signal is used.

Invert Select Bit

An invert select EPROM bit is used to invert the product term input into each macrocell register, including double inputs on JK and SR registers. This invert option allows the highest possible logic utilization by use of DeMorgan's logic inversion.

LOGIC ARRAY

Each intersecting point in the logic array contains a programmable EPROM connection. Initially (erased state), all connections are complete, i.e., both true and complement states of all signals are connected to each p-term.

Connections are opened during programming. When both the true and complement connections exist, a logical false results on the output of the AND gate. If both the true and complement connections of a signal are programmed "open", then a logic "don't care" results for that signal. If all connections for a p-term are programmed open, then a logical true results on the output of the AND gate.

PRODUCT TERM ALLOCATION

Product Term (p-term) allocation is defined as taking logic resources (p-terms) from macrocells where they are not used to support demand for additional p-terms in other macrocells. In the 5AC324, p-term allocation can occur in increments of 4 p-terms between adjacent macrocells. The 5AC324 includes 2 rings of 12 macrocells each. P-term groups from one macrocell can be allocated to the adjacent macrocell in the ring. P-term allocation between the two rings is not supported.

EXAMPLE:

Figure 5 shows a p-term allocation example. In this example, the logic function in macrocell 4 requires 16 p-terms. In this case, software allocates 4 p-terms from the previous macrocell in Ring 1 (macrocell 5) and 4 p-terms from the next macrocell (macrocell 3) to accumulate a total of 16 p-terms (8 + 4 + 4). This implementation leaves macrocells 3 and 5 with a remainder of 4 p-terms. These remaining p-terms can also be allocated away to, or supplemented with p-terms from, their adjacent macrocells in Ring 1 (macrocells 2 and 6).

With this scheme, any macrocell inside the device can support logic functions requiring between 0 and 16 p-terms. P-terms allocated away do not affect that macrocell's output structure. The input to the macrocell can be tied to VCC or GND, even when all p-terms have been allocated away. Thus the register and all control signals are still available for use if needed.

Figure 6 shows adjacent macrocells in the 5AC324. Table 1 shows the previous and next macrocells for each macrocell in the device, along with the corresponding allocation ring. P-term allocation is implemented automatically in the development software and is transparent to the user. Users can still use explicit pin assignment, but should assign pins in a way that does not conflict with p-term allocation.

Software support allows the control signals on macrocells to be used to implement simple logic functions even when all the input p-terms have been allocated to adjacent macrocells.

DUAL-FEEDBACK/BURIED LOGIC

Macrocell output can be fed back to the logic array on either one of the two feedback paths. If the pin feedback is used (connected after the output buffer), bidirectional I/O can be implemented. If the internal feedback path is used to implement a buried register or buried logic function, the pin feedback is still available for use as an input. The availability of dual feedbacks on the 5AC324 enhances resource efficiency over single feedback devices.

AUTOMATIC STAND-BY MODE

The 5AC324 contains a programmable bit, the Turbo Bit, that optimizes operation for speed or for power savings. When the Turbo Bit is programmed (TURBO = ON), the device is optimized for maximum

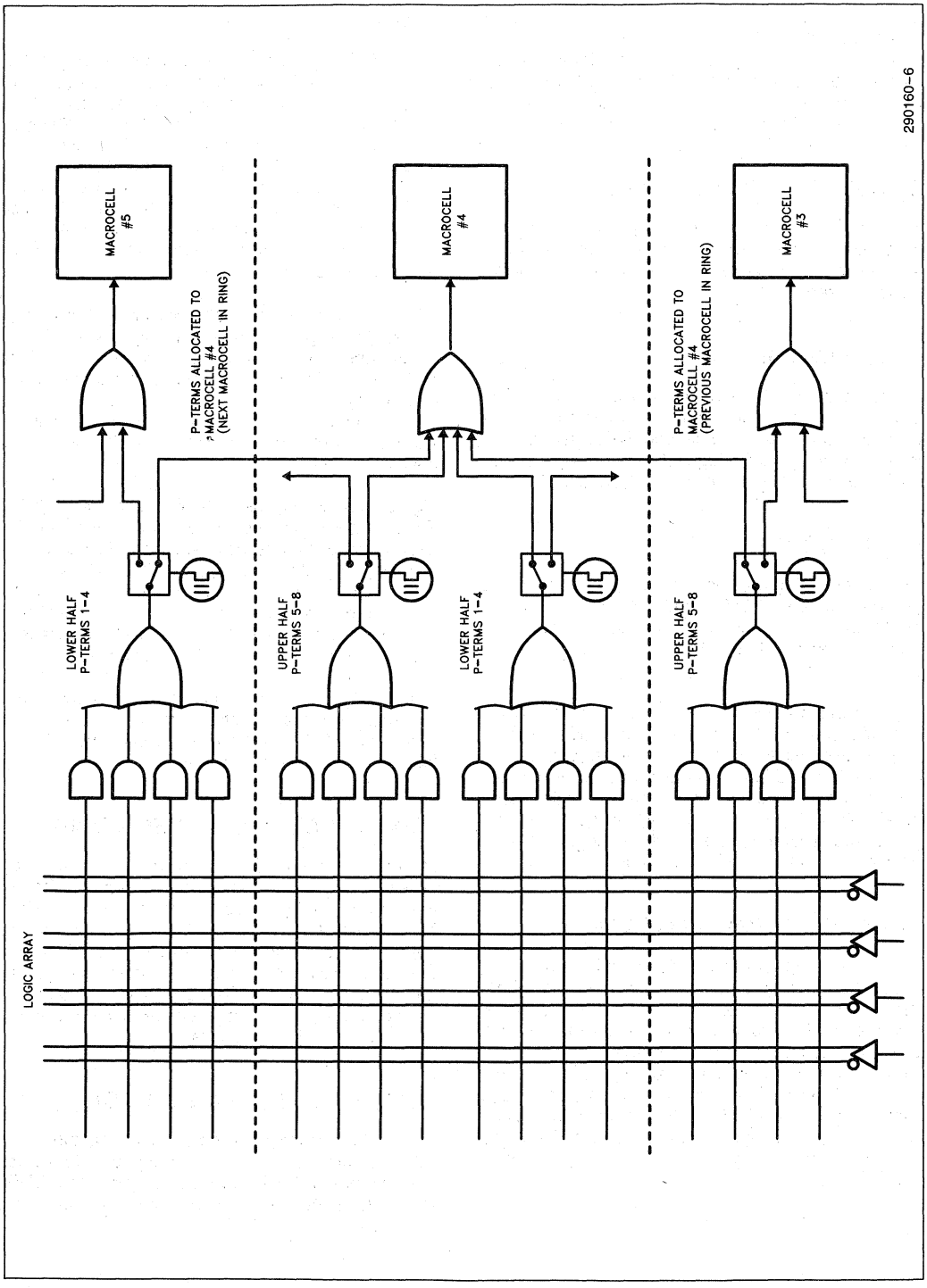


Figure 5. P-Term Allocation Example (8 + 4 + 4)

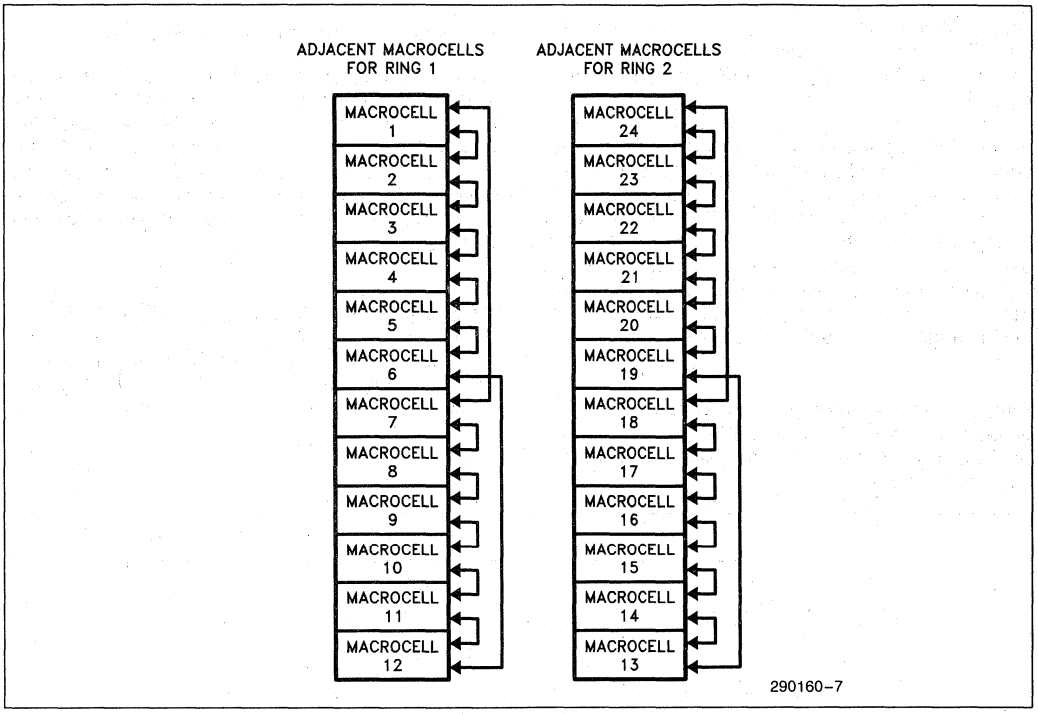


Figure 6. 5AC324 Adjacent Macrocell

Table 1. Product Term Allocation Rings

RING 1			RING 2		
Current Macrocell	**Next Macrocell**	**Previous Macrocell**	**Current Macrocell**	**Next Macrocell**	**Previous Macrocell**
1	7	2	13	19	14
2	1	3	14	13	15
3	2	4	15	14	16
4	3	5	16	15	17
5	4	6	17	16	18
6	5	12	18	17	24
7	8	1	19	20	13
8	9	7	20	21	19
9	10	8	21	22	20
10	11	9	22	23	21
11	12	10	23	24	22
12	6	11	24	18	23

speed. When the Turbo Bit is not programmed (TURBO = OFF), the device is optimized for power savings by entering standby mode during periods of inactivity.

Figure 7 shows the device entering standby mode approximately 100 ns after the last input or I/O transition. When the next input or I/O transition is detected, the device returns to active mode. Wakeup time adds an additional 20 ns to the propagation delay through the device as measured from the first transition. No delay will occur if an output is dependent on more than one input and the last of the inputs changes after the device has returned to active mode.

After erasure, the Turbo Bit is unprogrammed (OFF); automatic standby mode is enabled. When the Turbo Bit is programmed (ON), the device never enters standby mode.

POWER-ON CHARACTERISTICS

On V_{CC} power-up, the 5AC324 registers are reset to a logic low. Input latch/register output (to the logic array) are also set to a logic low. 5AC324 inputs and outputs begin responding approximately 20 μS after V_{CC} power-up or after a power-loss/power-up sequence. After power-up, macrocells can be preset to a logic high via the PRESET control signal for each macrocell.

ERASED STATE CONFIGURATION

After erasure and prior to programming, all macrocells are configured as combinatorial outputs with output buffers three-stated. Inputs are configured as synchronous registers.

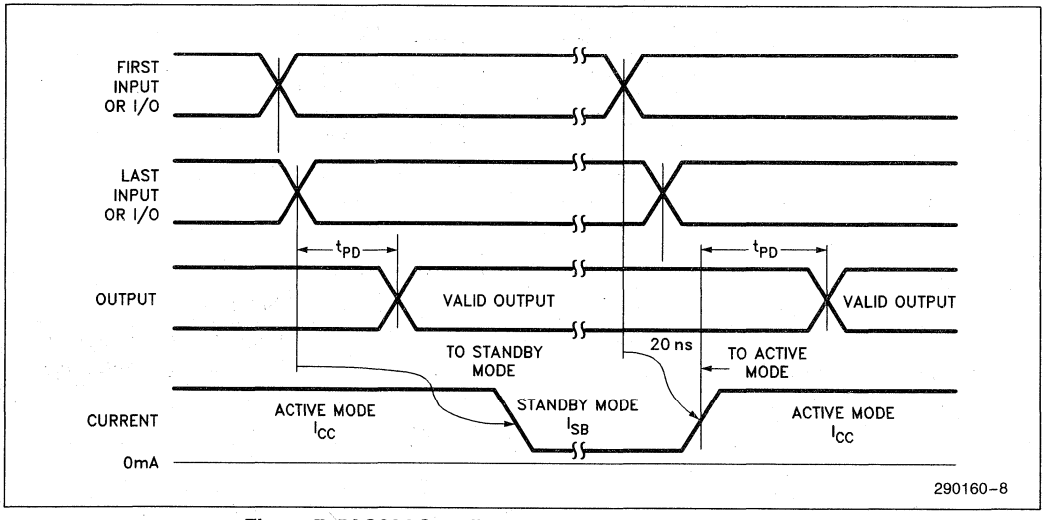


Figure 7. 5AC324 Standby and Active Mode Transitions

ERASURE CHARACTERISTICS

Erasure time for the 5AC324 is 1 hour at 12,000 $\mu W/cm^2$ with a 2537Å UV lamp.

Erasure characteristics of the device are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000Å. It should be noted that sunlight and certain types of flourescent lamps have wavelengths in the 3000Å-4000Å range. Data shows that constant exposure to room level flourescent lighting could erase the typical 5AC324 in approximately six years, while it would take approximately two weeks to erase the device when exposed to direct sunlight. If the device is to be exposed to these lighting conditions for extended periods of time, conductive opaque labels should be placed over the device window to prevent unintentional erasure.

The recommended erasure procedure for the 5AC324 is exposure to shortwave ultraviolet light with a wavelength of 2537Å. The integrated dose (i.e., UV intensity x exposure time) for erasure should be a minimum of forty (40) $Wsec/cm^2$. The erasure time with this dosage is approximately 1 hour using an ultraviolet lamp with a 12,000 $\mu W/cm^2$ power rating. The device should be placed within 1 inch of the lamp tubes during exposure. The maximum integrated dose the 5AC324 can be exposed to without damage is 7258 $Wsec/cm^2$ (1 week at 12,000 $\mu W/cm^2$). Exposure to high intensity UV light for longer periods may cause permanent damage to the device.

int$_e$ligent Programming™ Algorithm

The 5AC324 supports the int$_e$ligent Programming Algorithm, which rapidly programs Intel EPLDs, while maintaining a high degree of reliability. It is particularly suited for production programming environments. This method ensures reliability as the incremental programming margin of each bit has been verified during programming. Programming voltage and waveform specifications are available by request from Intel to support programming the device.

LATCH-UP IMMUNITY

All of the input, I/O, and clock pins of the device have been designed to resist latch-up which is inherent in inferior CMOS structures. The 5AC324 is designed with Intel's proprietary 1-micron CHMOS EPROM process. Thus, each of the pins will not experience latch-up with currents up to ±100 mA and voltages ranging from −0.5V to (V_{CC} + 0.5V). The programming pin is designed to resist latch-up to the 13.5V maximum device limit.

DESIGN RECOMMENDATIONS

For proper operation, it is recommended that all input and output pins be constrained to the voltage range GND ≤ (V_{IN} or V_{OUT}) ≤ V_{CC}. All unused inputs and I/Os should be tied high or low to minimize power consumption (do not leave them floating). A power supply decoupling capacitor of at least 0.2μF must be connected directly between each V_{CC} and GND pin.

As with all CMOS devices, ESD handling procedures should be used with the 5AC324 to prevent damage to the device during programming, assembly, and test.

FUNCTIONAL TESTING

Since the logical operation of the 5AC324 is controlled by EPROM elements, the device is completely testable during the manufacturing process. Each programmable EPROM bit controlling the internal logic is tested using application independent test patterns. EPROM cells in the device are 100% tested for programming and erasure. After testing, the devices are erased before shipments to the customers. No post-programming tests of the EPROM array are required.

The testability and reliability of EPROM-based programmable logic devices is an important feature over similar devices based on fuse technology. Fuse-based programmable logic devices require a user to perform post-programming tests to insure device functionality. During the manufacturing process, tests on fuse-based parts can only be performed in very restricted ways in order to avoid preprogramming the array.

ADF PRIMITIVES SUPPORTED

The following ADF primitives are supported by this device:

INP	NOTF
LINP	JOJF
RINP	JONF
CONF	SONF
COCF	SOSF
COIF	TOIF
RONF	TONF
ROIF	TOTF
RORF	CLKB
NOCF	LINB
NORF	
NOJF	
NOSF	

DESIGN SOFTWARE

Full logic compilation and programming support is provided by Version 2.0 (or later) of iPLS II. The GUPI 40D44J provides programming support for the device.

iPLS II software interfaces to several schematic capture packages to enable designs to be entered in schematic form. IPLDview-286/IPLDdraw allows the designer to use familiar TTL symbols or EPLD design primitive symbols. User-defined symbols are also supported. IPLDdraw also provides a path to A.C. timing simulation of EPLD designs.

SCHEMA III-PLD allows the designer to use TTL symbols, EPLD custom macros, or EPLD design primitive symbols. It also supports user-defined symbols.

Other design formats include Boolean equation entry (supported directly by iPLS II) and state machine entry (supported by iSTATE).

Refer to the iPLDS II (Intel Programmable Logic Development System II) Data Sheet (Order Number: 280168) for details on the software. Refer to the tools section of the *Programmable Logic* handbook for a complete listing of development tools.

The 5AC324 is also supported by third-party logic compilers such as ABEL†, CUPL†, PLDesigner†, Log/IC, etc. Programming support is provided by third-party programmer companies such as Data I/O, Logical Devices, STAG, etc. Please refer to the "Third-Party Support" lists in the *Programmable Logic* handbook for complete information and vendor contacts.

ORDERING INFORMATION

t_{PD} (ns)	t_{CO} (ns)	f_{MAX} (MHz)	Order Code	Package	Operating Range
25	17.8	66	N5AC324-25	PLCC	Commercial
			P5AC324-25	PDIP	
			D5AC324-25	*CERDIP	
30	20	50	N5AC324-30	PLCC	Commercial
			P5AC324-30	PDIP	
			D5AC324-30	*CERDIP	
35	22	40	N5AC324-35	PLCC	Commercial
			P5AC324-35	PDIP	
			D5AC324-35	*CERDIP	

*Windowed package allows UV erase.

†ABEL is a trademark of Data I/O Corporation. CUPL is a trademark of Logical Devices, Inc. PLDesigner is a trademark of MINC, Inc.

ABSOLUTE MAXIMUM RATINGS*

Supply Voltage (V_{CC})(1) ..... −2.0V to +7.0V

Programming Supply Voltage (V_{PP})(1) ..... −2.0V to +13.5V

D.C. Input Voltage (V_I)(1,2) .... −0.5V to V_{CC} +0.5V

Storage Temperature (T_{stg}) ..... −65°C to +150°C

Ambient Temperature (T_{amb})(3) ... −10°C to +85°C

NOTES:
1. Voltage with respect to GND.
2. Minimum D.C. input is −0.5V. During transitions, the inputs may undershoot to −2.0V or overshoot to +7.0V less than 20 ns under no load conditions.
3. Under bias. Extended Temperature versions are also available.

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

2

RECOMMENDED OPERATING CONDITIONS

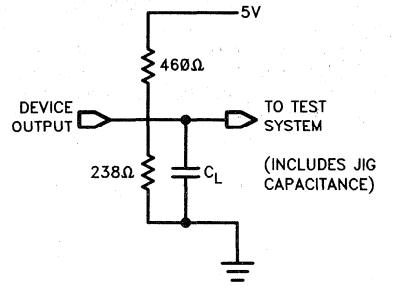
Symbol	Parameter	Min	Max	Unit
V_{CC}	Supply Voltage	4.75	5.25	V
V_{IN}	Input Voltage	0	V_{CC}	V
V_O	Output Voltage	0	V_{CC}	V
T_A	Operating Temperature	0	+70	°C
t_R	Input Rise Time		500	ns
t_F	Input Fall Time		500	ns

D.C. CHARACTERISTICS (T_A = 0°C to +70°C, V_{CC} = 5.0V ±5%)

Symbol	Parameter	Min	Typ	Max	Unit	Test Conditions
V_{IH}(4)	High Level Input Voltage	2.0		V_{CC} +0.3	V	
V_{IL}(4)	Low Level Input Voltage	−0.3		0.8		
V_{OH}(5)	High Level Output Voltage	2.4			V	I_O = −4.0 mA D.C., V_{CC} = min.
V_{OL}	Low Level Output Voltage			0.45	V	I_O = 8.0 mA D.C., V_{CC} = min.
I_I	Input Leakage Current			±10	µA	V_{CC} = max., GND < V_{IN} < V_{CC}
I_{OZ}	Output Leakage Current			±10	µA	V_{CC} = max., GND < V_{OUT} < V_{CC}
I_{SC}(6)	Output Short Circuit Current	−30		−90	mA	V_{CC} = max., V_{OUT} = 0.5V
I_{SB}(7)	Standby Current		150	500	µA	V_{CC} = max., V_{IN} = V_{CC} or GND, Standby Mode
I_{CC}	Power Supply Current (See I_{CC} vs Freq. Graph)		20		mA	V_{CC} = max., V_{IN} = V_{CC} or GND, No Load, f_{IN} = 1 MHz, Active Mode (Turbo = Off), Device Prog. as Two 12-Bit Counters

NOTES:
4. Absolute values with respect to device GND; all over and undershoots due to system or tester noise are included.
5. I_O at CMOS levels (3.84V) = −2 mA.
6. Not more than 1 output should be tested at a time. Duration of that test should not exceed 1 second.
7. With Turbo Bit Off, device automatically enters standby mode approximately 100 ns after last input transition.

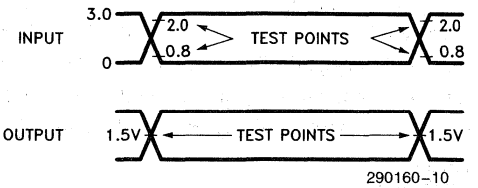
A.C. TESTING LOAD CIRCUIT



C_L = 30 pF

290160-9

A.C. TESTING INPUT, OUTPUT WAVEFORM



290160-10

A.C. Testing: Inputs are driven at 3.0V for a Logic "1" and 0V for a Logic "0". Timing Measurements are made at 2.0V for a Logic "1" and 0.8V for a Logic "0" on inputs. Outputs are measured at a 1.5V point. Device input rise and fall times < 6 ns.

CAPACITANCE

Symbol	Parameter	Min	Typ	Max	Unit	Conditions
C_{IN}	Input Capacitance			8	pF	V_{IN} = 0V, f = 1.0 MHz
C_{OUT}	I/O Capacitance			15	pF	V_{OUT} = 0V, f = 1.0 MHz
C_{CLK}	Clock Pin Capacitance			15	pF	V_{IN} = 0V, f = 1.0 MHz
C_{VPP}	V_{PP} Pin (LIN3)			25	pF	V_{IN} = 0V, f = 1.0 MHz

COMBINATORIAL MODE A.C. CHARACTERISTICS

(T_A = 0°C to +70°C, V_{CC} = 5.0V ±5%, Turbo Bit On)[8]

Symbol	Parameter	5AC324-25			5AC324-30			5AC324-35			Non-Turbo[9] Mode	Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
t_{PD}	Input or I/O to Output Valid		20	25		25	30		30	35	+20	ns
t_{PZX}[10]	Input or I/O to Output Enable		20	25		25	30		30	35	+20	ns
t_{PXZ}[10]	Input or I/O to Output Disable		20	25		25	30		30	35	+20	ns
t_{CLR}	Asynch. Reset to Q Reset		20	25		25	30		30	35	+20	ns
t_{SET}	Asynch. Set to Q Set		20	25		25	30		30	35	+20	ns

NOTES:
8. Typical values are at T_A = +25°C, V_{CC} = 5V, Active Mode.
9. If device is operated with Turbo bit Off (Non-Turbo Mode) and the device is inactive for approx. 100 ns, increase time by amount shown.
10. t_{PZX} and t_{PXZ} measured at ±0.5V from steady-state voltage as driven by spec. output load. t_{PXZ} measured with C_L = 5 pF.

SYNCHRONOUS CLOCK MODE (MACROCELLS) A.C. CHARACTERISTICS

(T_A = 0°C to +70°C, V_{CC} = 5.0V ±5%, Turbo Bit On)[8]

Symbol	Parameter	5AC324-25			5AC324-30			5AC324-35			Non-Turbo[9] Mode	Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
f_{MAX}	Maximum Frequency (Pipelined) ($1/t_{CW}$)—No Feedback		80	66		66	50		50	40		MHz
f_{CNT1}	Maximum Frequency ($1/t_{SU} + t_{CO}$) —External Feedback		40	33		33.3	25		27	21.2		MHz
f_{CNT2}	Maximum Frequency ($1/t_{CNT}$) —Internal Feedback		40	33.3		33.3	28.5		28.5	25		MHz
t_{SU1}	Input Setup Time to CLK ↑	12.5	10		20	15		25	20		+20	ns
t_{SU2}	I/O Setup Time to CLK ↑	12	10		20	15		25	20		+20	ns
t_H	Input or I/O Hold Time from CLK ↑	0			0			0				ns
t_{CO}	CLK ↑ to Output Valid		15	17.8		15	20		17	22		ns
t_{CNT}	Register Output Feedback to Register Input— Internal Path		25	30		30	35		35	40	+20	ns
t_{CH}	Clock High Time	7			9			11				ns
t_{CL}	Clock Low Time	7			9			11				ns
t_{CW}	Minimum Clock Period	15			20			25				ns

SYNCHRONOUS CLOCK MODE (INPUT STRUCTURE) A.C. CHARACTERISTICS

(T_A = 0°C to +70°C, V_{CC} = 5.0V ±5%, Turbo Bit On)[8]

Symbol	Parameter	5AC324-25			5AC324-30			5AC324-35			Non-Turbo[9] Mode	Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
f_{MAXI}	Maximum Frequency ($1/t_{CWI}$)		80	66		60	50		50	40		MHz
t_{SUIR}	Input Register Setup Time Before ILE/ICLK ↓	1			2.5			5				ns
t_{ESUI}[11]	Input Latch Setup Time Before ILE ↑	1			2.5			5				ns
t_{COI}	ICLK ↓ to Comb. Output		25	30		30	35		35	40	+20	ns
t_{EOI}	ILE ↑ to Comb. Output		25	30		30	35		35	40	+20	ns
t_{HI}	Input Hold after ICLK ↓	8			9			10				ns
t_{EHI}	Input Hold after ILE ↓	7			8			9				ns
t_{CHI}	ILE/ICLK High Time	7			9			11				ns
t_{CLI}	ILE/ICLK Low Time	7			9			11				ns
t_{CWI}	Minimum Input Clock Period	15			20			25				ns

ASYNCHRONOUS CLOCK MODE (MACROCELLS) A.C. CHARACTERISTICS

(T_A = 0°C to +70°C, V_{CC} = 5.0V ±5%, Turbo Bit On)[8]

Symbol	Parameter	5AC324-25			5AC324-30			5AC324-35			Non-Turbo[9] Mode	Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
f_{AMAX}	Max. Frequency (Pipelined) ($1/t_{ACW}$)—No Feedback		80	66		60	50		50	40		MHz
f_{ACNT_1}	Max. Frequency ($1/t_{ASU} + t_{ACO}$) External Feedback		32.2	27.7		27	23.8		22.2	20		MHz
f_{ACNT_2}	Max. Frequency ($1/t_{ACNT}$) Internal Feedback		40	33.3		33.3	28.5		28.5	25		MHz
t_{ASU1}	Input Setup Time to Asynch. CLK	11			12			15			+20	ns
t_{ASU2}	I/O Setup Time to Asynch. CLK	11			12			15			+20	ns
t_{AH}	Input or I/O Hold Time from Asynch. CLK	3	0		4	0		5	0			ns
t_{ACO}	Asynch. CLK to Output Valid		20	25		25	30		30	35	+20	ns
t_{ACNT}	Asynch. Output Feedback to Register Input - Internal Path		25	30		30	35		35	40	+20	ns
t_{ACH}	Asynch. CLK High Time	7			9			11			+20	ns
t_{ACL}	Asynch. CLK Low Time	7			9			11			+20	ns
t_{ACW}	Asynch. CLK Period	15			20			25			+20	ns

ASYNCHRONOUS CLOCK MODE (INPUT STRUCTURE) A.C. CHARACTERISTICS

(T_A = 0°C to +70°C, V_{CC} = 5.0V ±5%, Turbo Bit On)[8]

Symbol	Parameter	5AC324-25			5AC324-30			5AC324-35			Non-Turbo[9] Mode	Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
f_{AMAXI}	Maximum Frequency Input Register ($1/t_{ACWI}$)		80	66		60	50		50	40		MHz
t_{ASUIR}	Input Register Setup Time Before Asynch. ICLK	−5			−5			−5				ns
t_{AESUI}[11]	Input Latch Setup Time Before Asynch. ILE	−5			−5			−5				ns
t_{ACOI}	Asynch. ICLK to Comb. Output		25	30		30	35		45	50	+20	ns
t_{AEOI}	Asynch. ILE to Comb. Output		25	30		30	45		45	50	+20	ns
t_{AHI}	Input Hold after Asynch. ICLK	15			18			20				ns
t_{AEHI}	Input Hold after Asynch. ILE	14			17			19				ns
t_{ACHI}	Asynch. ILE/ICLK High Time	7			9			11			+20	ns
t_{ACLI}	Asynch. ILE/ICLK Low Time	7			9			11			+20	ns
t_{ACWI}	Minimum Input Clock Period	15			20			25			+20	ns

NOTE:

11. This specification must be met to guarantee t_{EOI}. When ILE goes high before data is valid, use t_{PD} instead of t_{EOI}.

INPUT-CLOCK-TO-MACROCELL-CLOCK A.C. CHARACTERISTICS

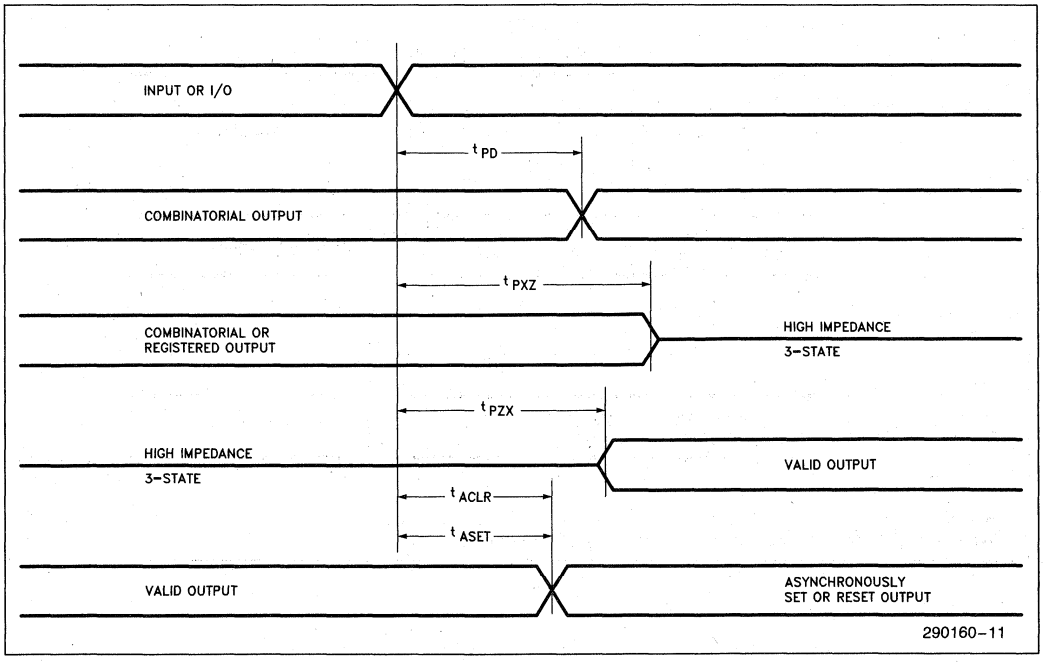
(T_A = 0°C to +70°C, V_{CC} = 5.0V ±5%, Turbo Bit On)[8]

Symbol	Parameter	5AC324-25			5AC324-30			5AC324-35			Non-Turbo[9] Mode	Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
t_{C1C2}[12]	Synchronous ILE/ICLK to Synchronous Macrocell CLK	20			25			30			+20	ns
	Synchronous ILE/ICLK to Asynchronous Macrocell CLK	12.5			15			18			+20	ns
	Asynchronous ILE/ICLK to Synchronous Macrocell CLK	40			45			50			+20	ns
	Asynchronous ILE/CLK to Asynchronous Macrocell CLK	20			25			30			+20	ns

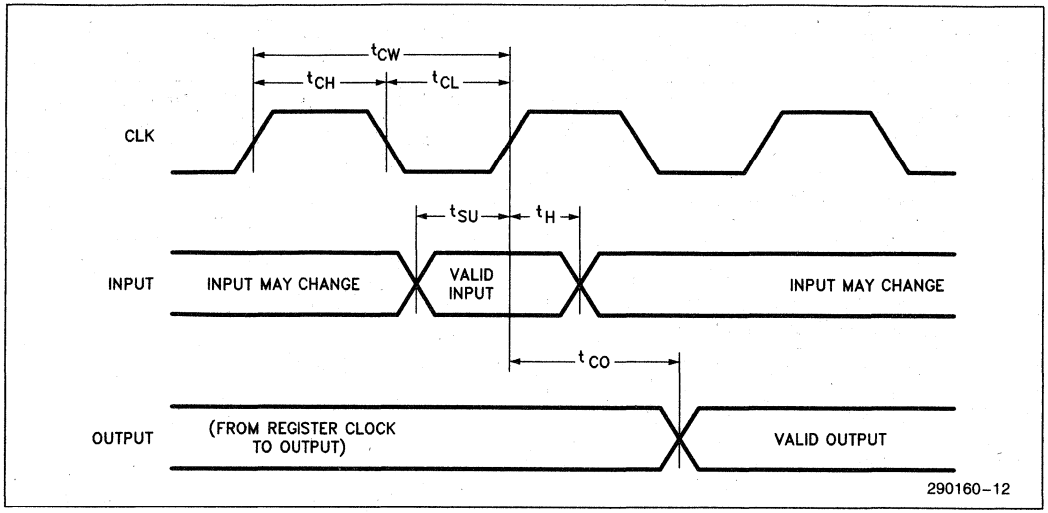
NOTE:
12. Times for SETUP, HOLD, and OUTPUT VALID are shown in previous tables.

SWITCHING WAVEFORMS

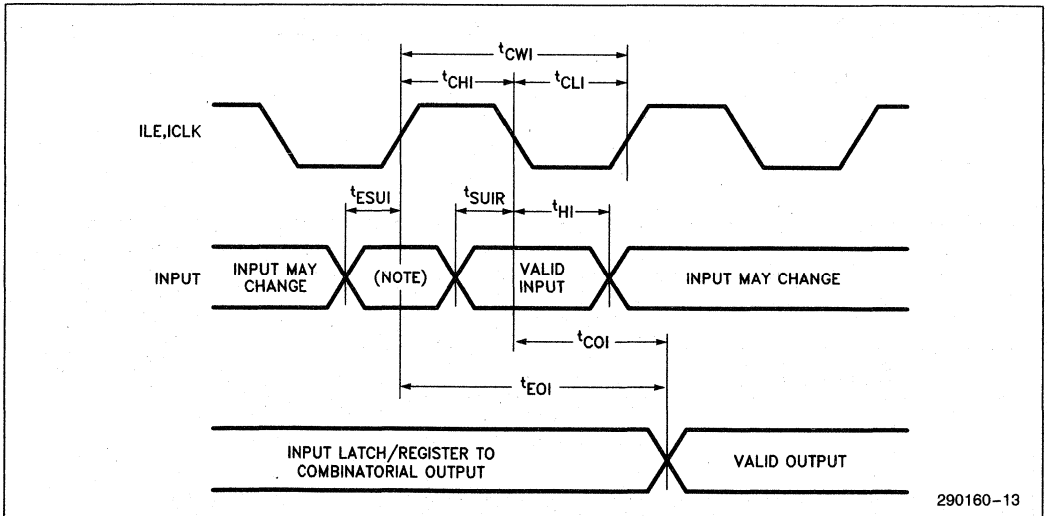
COMBINATORIAL MODE



SYNCHRONOUS CLOCK MODE (MACROCELLS)

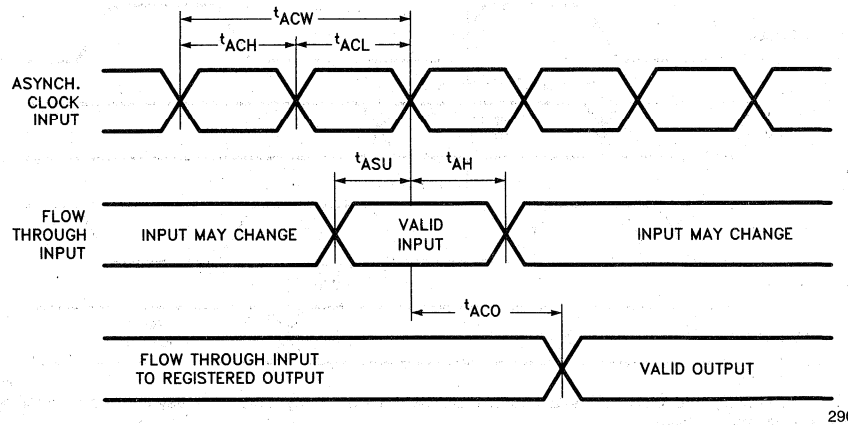


SYNCHRONOUS CLOCK MODE (INPUT STRUCTURE)

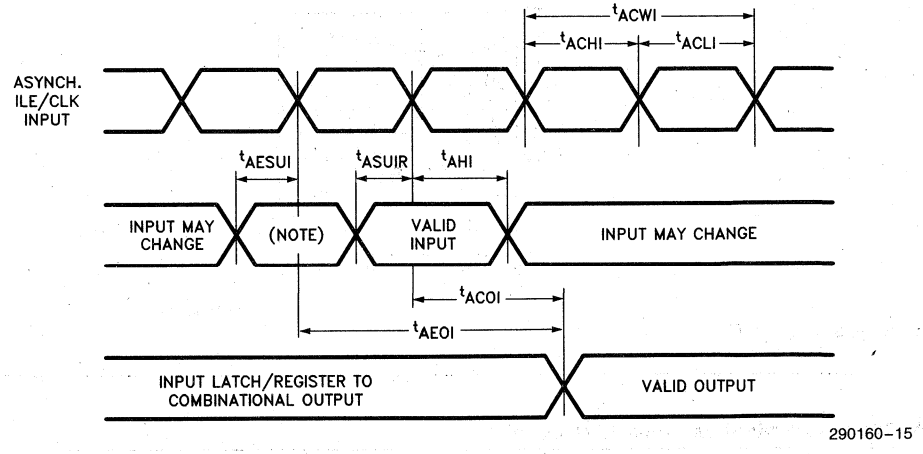


NOTE:
When ILE goes high before data is valid, use t_{PD} instead of t_{EOI}.

ASYNCHRONOUS CLOCK MODE (MACROCELLS)

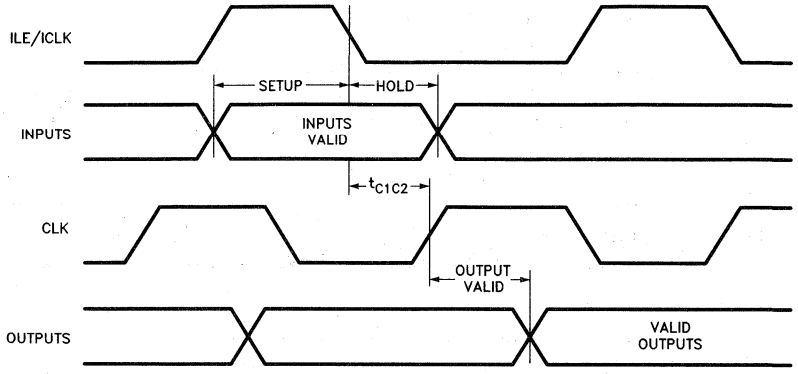


ASYNCHRONOUS CLOCK MODE (INPUT STRUCTURE)



NOTE:
When ILE goes high before data is valid, use t_{PD} instead of t_{AEOf}.

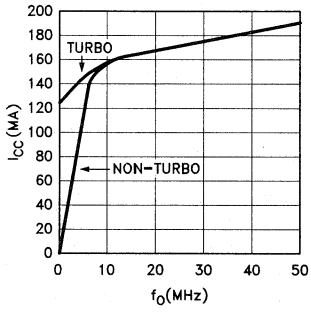
INPUT-CLOCK-TO-MACROCELL CLOCK TIMING (CLOCKED PIPELINED DATA)



290160-16

CLOCK, SETUP, HOLD, and OUTPUT VALID times are dependent on synchronous/asynchronous clocking and are listed in the specification tables.

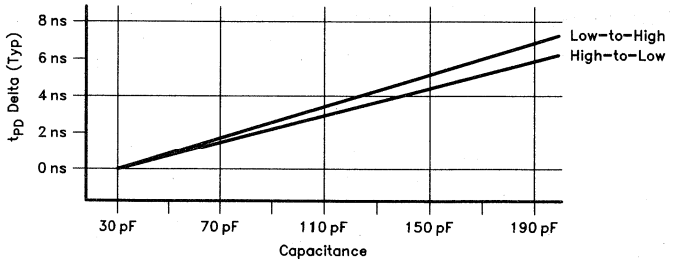
5AC324 I_{CC} vs. Frequency



290160-17

Conditions: T_A = 25°C
V_{CC} = 5.0V

5AC324 t_{PD} Derating vs. Capacitive Loading



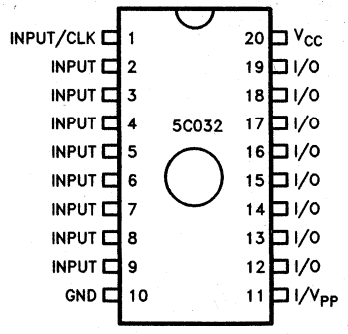
290160-18

Conditions: T_A = 25°C
V_{CC} = 5.0V

5C032
8-MACROCELL CHMOS EPLD

- High-Density, Low-Power Replacement for SSI & MSI Devices and Bipolar PLDs
- Up to 18 Inputs (10 Dedicated & 8 I/O) and 8 Outputs
- Eight Macrocells with Programmable I/O Architecture
- t_{PD} = 30 ns (max), 43.5 MHz Pipelined, 28.5 MHz with Feedback
- Low Power Upgrade for All Commonly Used 20-pin PLDs
- CHMOS EPROM Technology Based UV Erasable (CerDIP)
- Programmable "Security Bit" Allows Total Protection of Proprietary Designs
- I_{CC} (standby) 100 μA (max)
  I_{CC} (10 MHz) 25 mA (max)
- 100% Generically Tested EPROM Logic Control Array
- 20-pin 0.3" Ceramic and Plastic DIP Package
  (See Packaging Spec., Order #231369)
- 100% Compatible with EP320

2



290155-1

Pin Configuration

September 1990
Order Number: 290155-005

The Intel 5C032 is an 8-macrocell, 20-pin, general-purpose EPLD (Erasable Programmable Logic Device).This device can be used to replace bipolar programmable logic arrays and LS TTL and 74HC (CMOS) SSI and MSI logic devices. The 5C032 can also be used as a direct, low-power replacement for almost all common 20-pin fuse-based programmable logic devices. With its flexible programmable I/O architecture, this device is a superset of common 20-pin PLDs.

The 5C032 EPLD uses CHMOS EPROM (floating gate) cells as logic control elements instead of fuses. The CHMOS EPROM technology reduces power consumption of EPLDs to less than 20% of a comparable bipolar device without sacrificing speed performance. In addition, the use of Intel's advanced CHMOS II-E EPROM process technology enables greater logic densities to be achieved with superior speed and low-power performance over other comparable devices. Intel's 5C032 has the benefit of "zero" stand-by power not available on other programmable logic devices. EPROM technology allows these devices to be 100% factory tested by programming and erasing all the EPROM logic control elements.

The 5C032 with its superior speed and power performance and its plastic package is an ideal production vehicle for high-volume manufacturing. Most commonly used 20-pin bipolar PLDs can be easily replaced with this device allowing for tremendous power consumption savings without sacrificing speed of operation.

ARCHITECTURE DESCRIPTION

The architecture of the 5C032 is based on the "Sum of Products" PLA (Programmable Logic Array) structure with a programmable AND array feeding into a fixed OR array. This device can accommodate both combinational and sequential logic functions. A proprietary programmable I/O architecture provides individual selection of either combinational or registered output and feedback signals, all with selectable polarity.

The 5C032 contains 10 dedicated inputs as well as 8 input/output pins. These I/O pins can be individually configured to be inputs, outputs or bi-directional I/O pins. Each of these I/O pins is connected to a macrocell. The 5C032 contains 8 identical macrocells organized as shown in Figure 1.

Each macrocell (see Figure 2) consists of a PLA (programmable logic array) block and an I/O architecture block, which contains a "D" type register. The PLA block consists of eight 36-input AND gates (TRUE & COMPLEMENT of 10 dedicated inputs plus the 8 feedback inputs from the eight macrocells), feeding into an OR gate. The output of this PLA block is fed into the I/O architecture block. The different I/O and feedback options that are available in the 5C032 I/O block are shown in Figure 3.

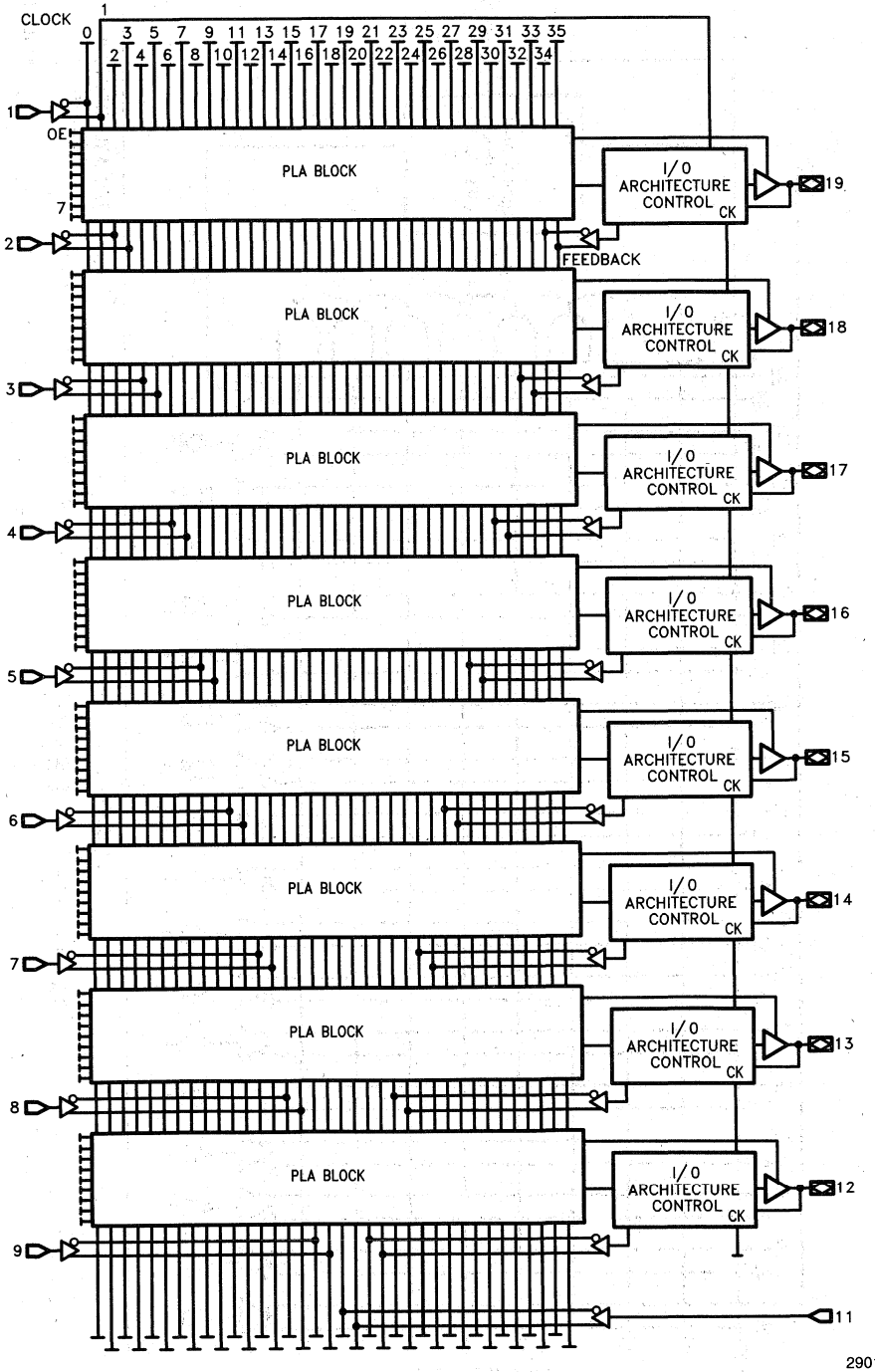


Figure 1. 5C032 Architecture

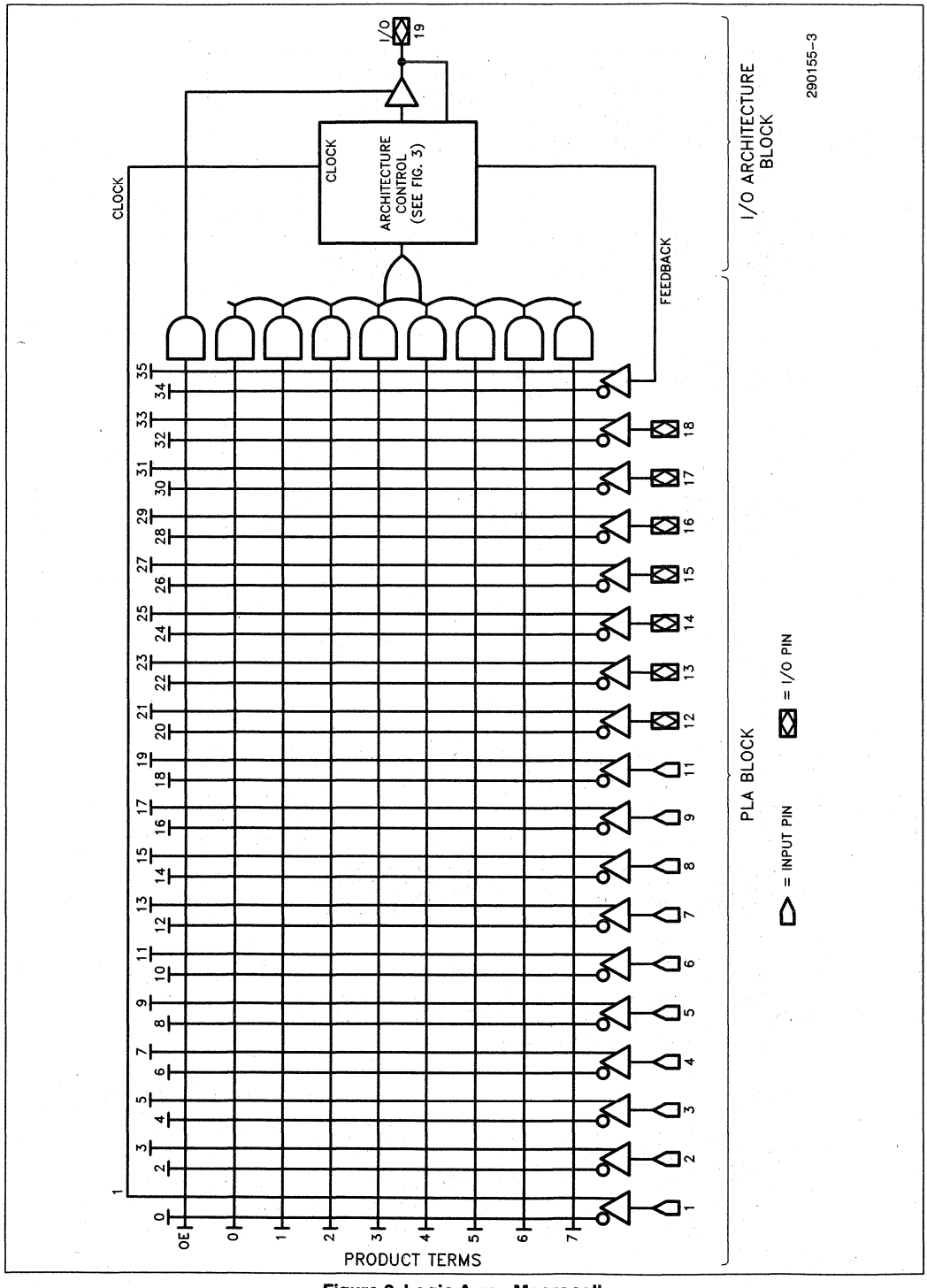


Figure 2. Logic Array Macrocell

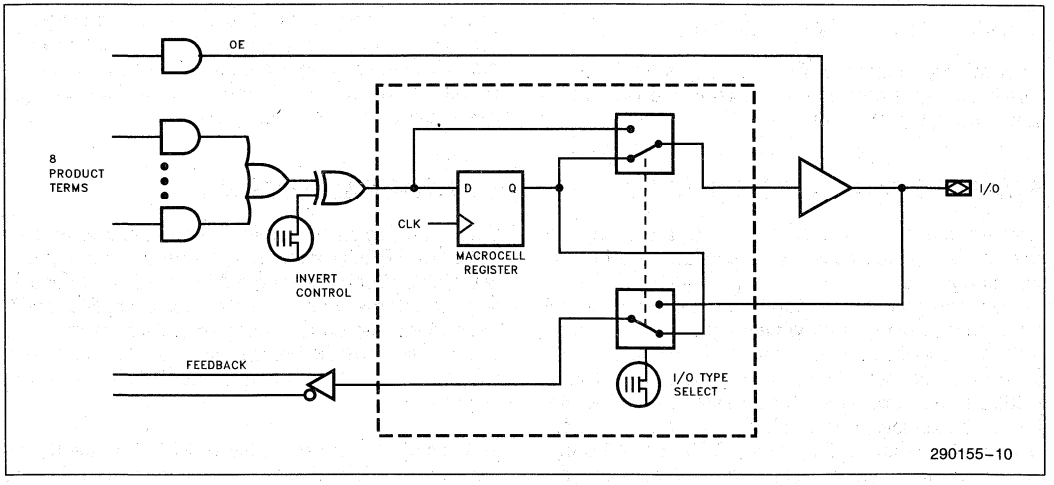


Figure 3. 5C032 I/O Architecture Control

20 PIN CMOS COMPATIBILITY

The 5C032 is architected to be a logical superset of most 20 pin bipolar programmable array logic (PAL*) devices. The I/O and logic sections of the 5C032 device can be configured to emulate any of the devices listed below. Designers can make use of this feature by reducing the power of PAL based systems (EPLDs are much lower power), replacing multiple PAL inventory items with a single EPLD. Designers can also create new 20 pin PLD configurations by utilizing the individual logic and output controls of each macrocell.

List of PAL devices logically compatible with the 5C032.

16V8	16L2
10H8	16L8
12H6	16R8
14H4	16R6
16H2	16R4
16H8	16P8A
16C1	16RP8A
10LB	16RP6A
12L6	16RP4A
14L4	

*PAL is a registered trademark of Advanced Micro Devices.

Erased-State Configuration

Prior to programming or after erasing, the I/O structure is configured for combinatorial active low output with input (pin) feedback.

ERASURE CHARACTERISTICS

Erasure characteristics of the 5C032 are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000Å. It should be noted that sunlight and certain types of flourescent lamps have wavelengths in the 3000–4000Å. Data shows that constant exposure to room level flourescent lighting could erase the typical 5C032 in approximately three years, while it would take approximately one week to cause erasure when exposed to direct sunlight. If the 5C032 is to be exposed to these types of lighting conditions for extended periods of time, conductive opaque labels should be placed over the device window to prevent unintentional erasure.

The recommended erasure procedure for the 5C032 is exposure to shortwave ultraviolet light with a wavelength of 2537Å. The integrated dose (i.e., UV intensity $\times$ exposure time) for erasure should be a minimum of fifteen (15) Wsec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12,000 μW/cm^2 power rating. The 5C032 should be placed within one inch of the lamp tubes during erasure. The maximum integrated dose the 5C032 can be exposed to without damage is 7258 Wsec/cm^2 (1 week at 12,000 μW/cm^2). Exposure to high intensity UV light for longer periods may cause permanent damage to the device.

PROGRAMMING CHARACTERISTICS

Initially, and after erasure, all the EPROM control bits of the 5C032 are connected (in the "1" state). Each of the connected control bits are selectively disconnected by programming the EPROM cells into their "0" state. Programming voltage and waveform specifications are available by request from Intel to support programming of the device.

int$_e$ligent Programming™ Algorithm

The 5C032 supports the int$_e$ligent Programming Algorithm which rapidly programs Intel H-ELPDs (and EPROMs) using an efficient and reliable method. The int$_e$ligent Programming Algorithm is particularly suited to the production programming environment. This method greatly decreases the overall programming time while programming reliability is ensured as the incremental program margin of each bit is continually monitored to determine when the bit has been successfully programmed.

FUNCTIONAL TESTING

Since the logical operation of the 5C032 is controlled by EPROM elements, the device is completely testable. Each programmable EPROM bit controlling the internal logic is tested using application-independent test program patterns. After testing, the devices are erased before shipment to customers. No post-programming tests of the EPROM array are required.

The testability and reliability of EPROM-based programmable logic devices is an important feature over similar devices based on fuse technology. Fuse-based programmable logic devices require a user to perform post-programming tests to insure proper programming. These tests must be done at the device level because of the cummulative error effect. For example, a board containing ten devices each possessing a 2% device fallout translates into an 18% fallout at the board level (it should be noted that programming fallout of fuse-based programmable logic devices is typically 2% or higher).

DESIGN RECOMMENDATIONS

For proper operation, it is recommended that all input and output pins be constrained to the voltage range GND $<$ (V_{IN} or V_{OUT}) $<$ V_{CC}. Unused inputs should be tied to an appropriate logic level (e.g. either V_{CC} or GND) to minimize device power consumption. Reserved pins (as indicated in the iPLDS REPORT file) should be left floating (no connect) so that the pin can attain the appropriate logic level. A power supply decoupling capacitor of at least 0.2 μF must be connected directly between V_{CC} and GND pins of the device.

As with all CMOS devices, ESD handling procedures should be used with the 5C032 to prevent damage to the device during programming, assembly, and test.

DESIGN SECURITY

A single EPROM bit provides a programmable design security feature that controls the access to the data programmed into the device. If this bit is set, a proprietary design within the device cannot be copied. This EPROM security bit enables a higher degree of design security than fused-based devices since programmed data within EPROM cells is invisi-

ble even to microscopic evaluation. The EPROM security bit, along with all the other EPROM control bits, will be reset by erasing the device.

AUTOMATIC STAND-BY MODE

The 5C032 contains a programmable bit, the Turbo Bit, that optimizes operation for speed or for power savings. When the Turbo Bit is programmed (TURBO = ON), the device is optimized for maximum speed. When the Turbo bit is not programmed (TURBO = OFF), the device is optimized for power savings by entering standby mode during periods of inactivity.

Figure 4 shows the device entering standby mode approximately 100 ns after the last input transition. When the next input transition is detected, the device returns to active mode. Wakeup time adds an additional 15 ns to the propagation delay through the device as measured from the first input. No delay will occur if an output is dependent on more than one input and the last of the inputs changes after the device has returned to active mode.

After erasure, the Turbo Bit is unprogrammed (OFF); automatic standby mode is enabled. When the Turbo Bit is programmed (ON), the device never enters standby mode.

LATCH-UP IMMUNITY

All of the input, I/O, and clock pins of the 5C032 have been designed to resist latch-up which is inherent in inferior CMOS structures. The 5C032 is designed with Intel's proprietary CHMOS II-E EPROM process. Thus, each of the 5C032 pins will not experience latch-up with currents up to ±100 mA and voltages ranging from −1V to (V_{CC} + 1V). Furthermore, the programming pin is designed to resist latch-up to the 13.5V maximum device limit.

INTEL PROGRAMMABLE LOGIC DEVELOPMENT SYSTEM II (iPLDS II)

iPLDS II provides all the tools needed to design with Intel EPLDs or compatible devices. In addition to providing development assistance, iPLDS II insulates the user from having to know all the intricate details of EPLD architecture (the machine will optimize a design to benefit from architectual features). It contains comprehensive third generation software that supports four different design entry methods, minimizes logic, does automatic pin assignments and produces the best design fit for the selected EPLD. It is user friendly with guided menus, on-line Help messages and soft key inputs.

In addition, iPLDS II contains programmer hardware in the form of an iUP-PC Universal Programmer-Personal Computer to enable the user to program EPLDs, read and verify programmed devices and also to graphically edit programming files. The software generates industry standard JEDEC object code output files which can be downloaded to other programmers as well.

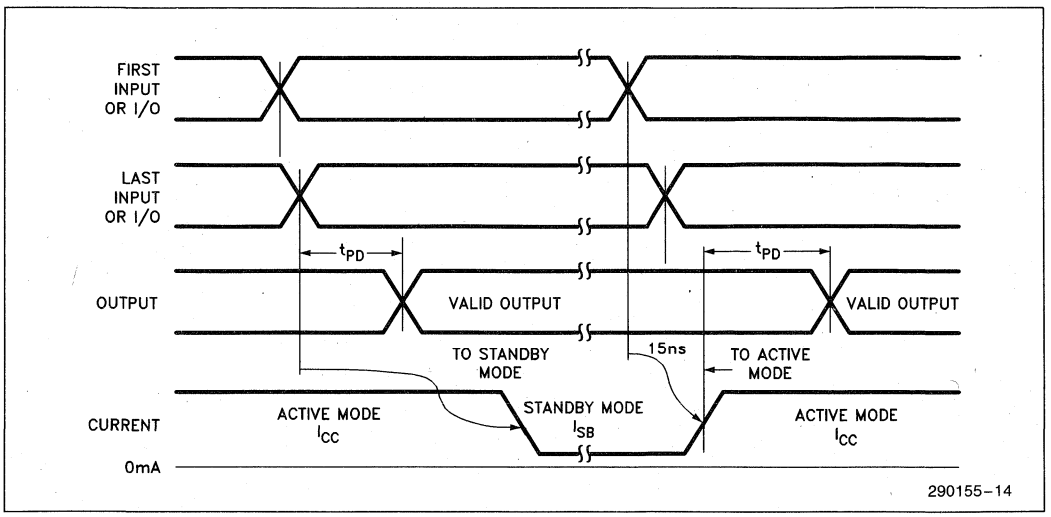


Figure 4. 5C032 Standby and Active Mode Transitions

iPLS II software interfaces to several schematic capture packages to enable designs to be entered in schematic form. IPLDview-286/IPLDdraw allows the designer to use familiar TTL symbols or EPLD design primitive symbols. User-defined symbols are also supported. IPLDdraw also provides a path to A.C. timing simulation of EPLD designs.

SCHEMA, III-PLD allows the designer to use TTL symbols, EPLD custom macros, or EPLD design primitive symbols. It also supports user-defined symbols.

Detailed information on the Intel Programmable Logic Development System II is contained in a separate Intel data sheet. (Order Number: 280168.) The tools section of the *Programmable Logic* handbook contains a complete listing of PLD design tools.

The 5C032 is also supported by third-party compilers such as ABEL*, CUPL*, PLDesigner*, Log/IC, etc. Programming support is provided by third-party programmer companies such as Data I/O, Logical Devices, STAG, etc. Please refer to the "Third-Party Support" lists in the *Programmable Logic* handbook for complete information and vendor contacts.

ADF PRIMITIVES SUPPORTED

The following ADF primitives are supported by this device:

INP	RONF
CONF	RORF
COIF	NORF

ORDERING INFORMATION

t_{PD} (ns)	t_{CO} (ns)	f_{MAX} (MHz)	Order Code	Package	Operating Range
30	17	43.5	D5C032-30	CERDIP	Commercial
			P5C032-30	PDIP	
35	20	40	D5C032-35	CERDIP	Commercial
			P5C032-35	PDIP	
40	24	33.3	D5C032-40	CERDIP	Commercial
			P5C032-40	PDIP	

*ABEL is a trademark of Data I/O, Corp. CUPL is a trademark of Logical Devices, Inc. PLDesigner is a trademark of MINC, Inc.

ABSOLUTE MAXIMUM RATINGS*

Symbol	Parameter	Min	Max	Unit
V_{CC}	Supply Voltage(1)	−2.0	7.0	V
V_{PP}	Programming Supply Voltage(1)	−2.0	13.5	V
V_I	DC Input Voltage(1)(2)	−0.5	V_{CC}+0.5	V
t_{stg}	Storage Temperature	−65	+150	°C
t_{amb}	Ambient Temperature(4)	−10	+85	°C

NOTES:
1. Voltages with respect to ground.
2. Minimum DC input is −0.5V. During transitions, the inputs may undershoot to −2.0V or overshoot to +7.0V for periods less than 20 ns under no load conditions.
3. Under bias, Extended temperature versions are also available.
4. Extended temperature versions also available.

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

2

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Min	Max	Unit
V_{CC}	Supply Voltage	4.75	5.25	V
V_{IN}	Input Voltage	0	V_{CC}	V
V_O	Output Voltage	0	V_{CC}	V
T_A	Operating Temperature	0	+70	°C
t_R	Input Rise Time		500	ns
t_F	Input Fall Time		500	ns

D.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ±5%

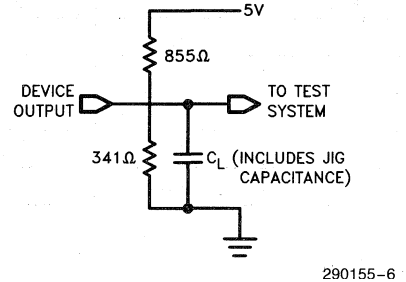
Symbol	Parameter/Test Conditions	Min	Typ	Max	Unit
V_{IH}(5)	High Level Input Voltage	2.0		V_{CC} + 0.3	V
V_{IL}(5)	Low Level Input Voltage	−0.3		0.8	V
V_{OH}(6)	High Level Output Voltage I_O = −4.0 mA D.C., V_{CC} = min.	2.4			V
V_{OL}	Low Level Output Voltage I_O = 4.0 mA D.C., V_{CC} = min.			0.45	V
I_I	Input Leakage Current V_{CC} = max., GND < V_{IN} < V_{CC}			±10	μA
I_{OZ}	Output Leakage Current V_{CC} = max., GND < V_{OUT} < V_{CC}			±10	μA

D.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ±5% (Continued)

Symbol	Parameter/Test Conditions	Min	Typ	Max	Unit
I_{SC}(7)	Output Short Circuit Current V_{CC} = max., V_{OUT} = 0.5V			10	mA
I_{SB}(8)	Standby Current V_{CC} = max., V_{IN} = V_{CC} or GND, Standby Mode		10	100	µA
I_{CC}(9)	Power Supply Current V_{CC} = max., V_{IN} = V_{CC} or GND, No Load, Input Freq. = 10 MHz Active Mode (Turbo = Off), Device Prog. as 8-bit Ctr.		15	25	mA

NOTES:
5. Absolute values with respect to device GND; all over- and undershoots due to system or tester noise are included.
6. I_O at CMOS levels (3.84V) = −2 mA.
7. Not more than 1 output should be tested at a time. Duration of that test must not exceed 1 second.
8. With Turbo Bit = Off, device automatically enters standby mode approximately 100 ns after last input transition.
9. Maximum Active Current at operational frequency is less than 40 mA.

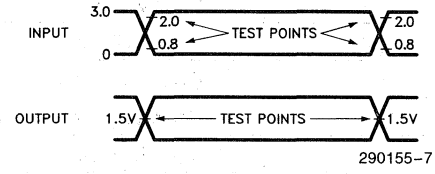
A.C. TESTING LOAD CIRCUIT



290155-6

C_L = 50 pF

A.C. TESTING INPUT, OUTPUT WAVEFORM



290155-7

A.C. Testing: Inputs are Driven at 3.0V for a Logic "1" and 0V for a Logic "0". Timing Measurements are made at 2.0V for a Logic "1" and 0.8V for a Logic "0" on inputs. Outputs are measured at a 1.5V point. Device input rise and fall times < 6 ns.

CAPACITANCE

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
C_{IN}	Input Capacitance	V_{IN} = 0V, f = 1.0 MHz			10	pF
C_{OUT}	Output Capacitance	V_{OUT} = 0V, f = 1.0 MHz			10	pF
C_{CLK}	Clock Pin Capacitance	V_{IN} = 0V, f = 1.0 MHz			10	pF
C_{VPP}	V_{PP} Pin	Pin 11, f = 1.0 MHz			20	pF

A.C. CHARACTERISTICS T_A = 0°C to +70°C, V_{CC} = 5V ±5%, Turbo Bit On[10]

Symbol	From	To	5C032-30			5C032-35			5C032-40			Non-[8] Turbo Mode	Unit
			Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
t_{PD}	I or I/O	Comb. Output			30			35			40	+15	ns
t_{PZX}[11]	I or I/O	Output Enable			30			35			40	+15	ns
t_{PXZ}[11]	I or I/O	Output Disable			30			35			40	+15	ns

NOTES:
10. Typ. values are at T_A = 25°C, V_{CC} = 5V, Active Mode.
11. t_{PZX} and t_{PXZ} are measured at ±0.5V from steady state voltage as driven by spec. output load. t_{PXZ} is measured with C_L = 5 pF.

A.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ±5%, Turbo Bit On [10]

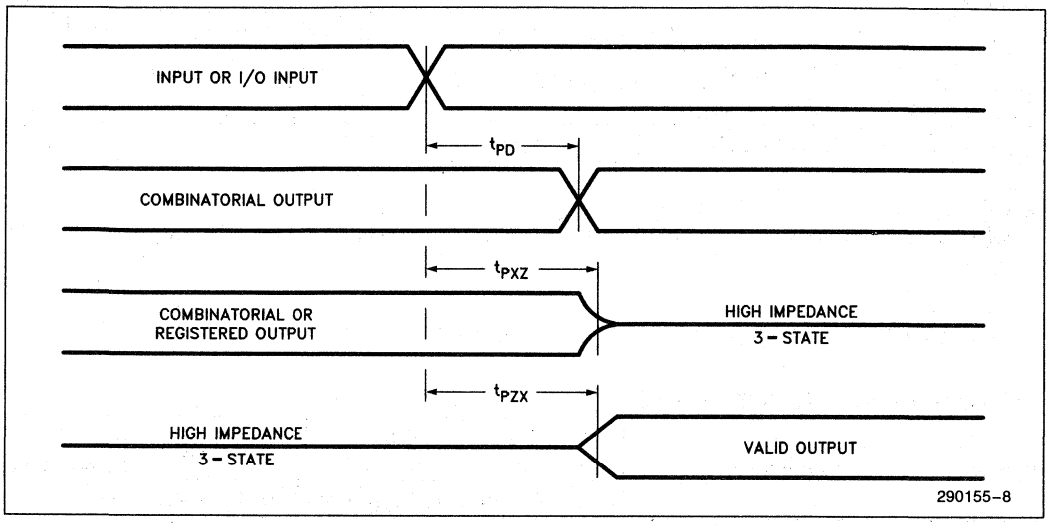
2

SYNCHRONOUS CLOCK MODE

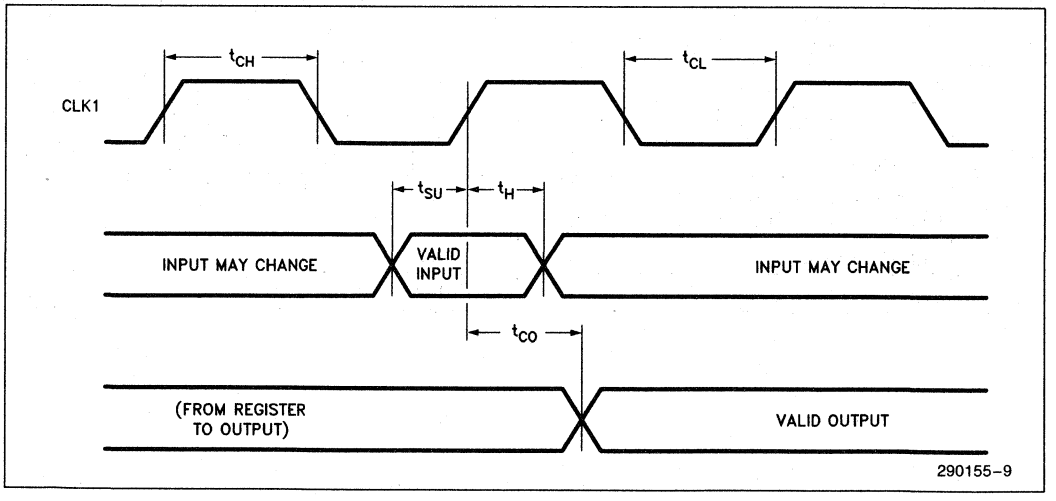
Symbol	Parameter	5C032-30 EP320-1			5C032-35 EP320-2			5C032-40			Non-[8] Turbo Mode	Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
f_{MAX}	Max. Frequency (Pipelined) $1/t_{SU}$ — No Feedback			43.5			40			33.3		MHz
f_{CNT}	Max. Count Frequency $1/t_{CNT}$ — with Internal Feedback			28.5			25			21.7		MHz
t_{SU}	Input Setup Time to CLK	23			25			30			+15	ns
t_H	I or I/O Hold after CLK High	0			0			0				ns
t_{CO}	CLK High to Output Valid			17			20			24		ns
t_{CNT}	Register Output Feedback to Register Input — Internal Path	35			40			46			+15	ns
t_{CH}	CLK High Time	11			12			15				ns
t_{CL}	CLK Low Time	11			12			15				ns

SWITCHING WAVEFORMS

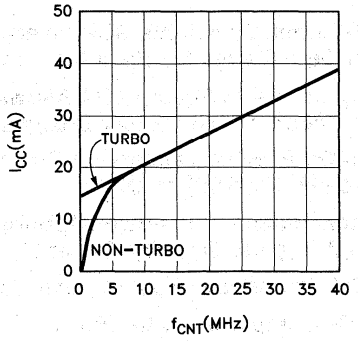
COMBINATORIAL MODE



SYNCHRONOUS CLOCK MODE



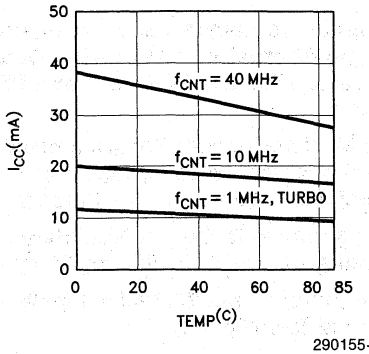
Current in Relation to Frequency



290155-11

Conditions: T_A = 0°C, V_{CC} = 5.25V

Current in Relation to Temperature

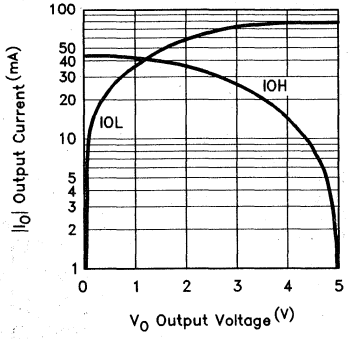


290155-12

Conditions: V_{CC} = 5.25V

2

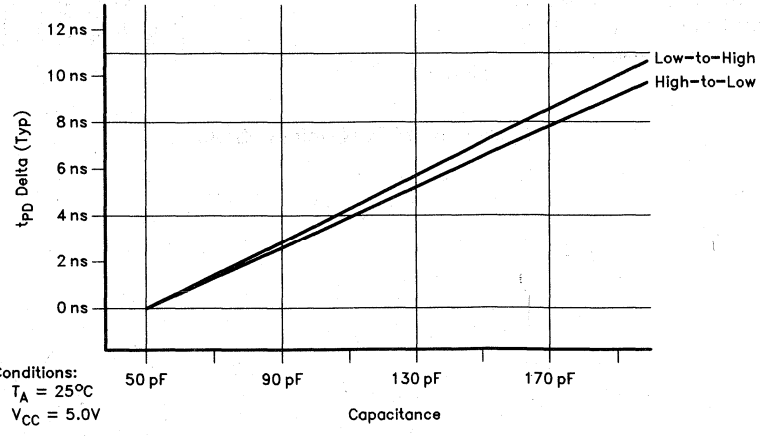
Output Drive Current in Relation to Voltage



290155-13

Conditions: T_A = +25°C, V_{CC} = 5V

t_{PD} Derating vs Capacitive Loading

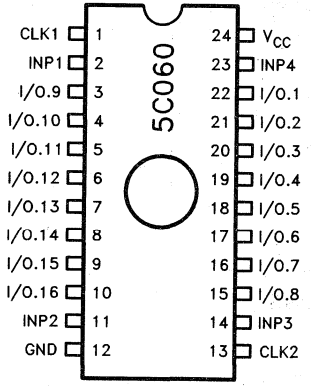


Conditions:
T_A = 25°C
V_{CC} = 5.0V

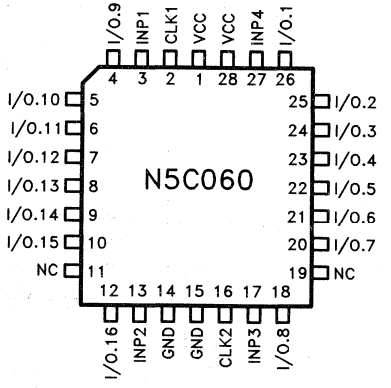
290155-15

5C060
16-MACROCELL CHMOS EPLD

- High-Performance LSI Semi-Custom Logic Alternative to Low-End Gate Arrays, TTL, and 74HC SSI and MSI Logic
- 16 Macrocells with Programmable I/O Architecture; up to 20 Inputs (4 Dedicated, 16 I/O) or 16 Outputs
- Programmable Output Registers can be Configured as D, T, SR, or JK Types
- t_{PD} (max) 45 ns, 26.3 MHz Pipelined, 22.2 MHz w/Feedback
- Programmable Clock System with 2 Synchronous Clocks and Asynchronous Clocking Option on all Registers
- 8 P-Terms, Selectable SOP Invert, Clear and OE P-Terms for Each Macrocell
- Programmable Security Bit Allows Total Protection of Proprietary Designs
- CHMOS EPROM Technology Based. UV Erasable (CerDIP) or OTP
- Progammable Low Power Option; 50 μA Typical Standby Current
- 100% Generically Tested Logic Array
- 100% Compatible with EP600
- Available in 24-Pin 300-mil CerDIP/PDIP and 28-Pin PLCC Packages
  (See Packaging Spec. Order #231369)



290194-1



290194-2

Figure 1. 5C060 Pin Configurations

September 1990
Order Number: 290194-003

The Intel 5C060 EPLD (Erasable Programmable Logic Device) is a 16-macrocell, 24-pin, general purpose device. The device can be used to replace low-end gate arrays, multiple programmable logic arrays and LS TTL and 74HC (CMOS) SSI and MSI logic devices. The 5C060 can also be used as a direct, low-power replacement for most, common 24-pin fuse-based programmable logic devices. With its revolutionary programmable I/O architecture, the device has advanced functional capabilities beyond that of typical programmable logic. Figure 2 shows the global architecture of the device.

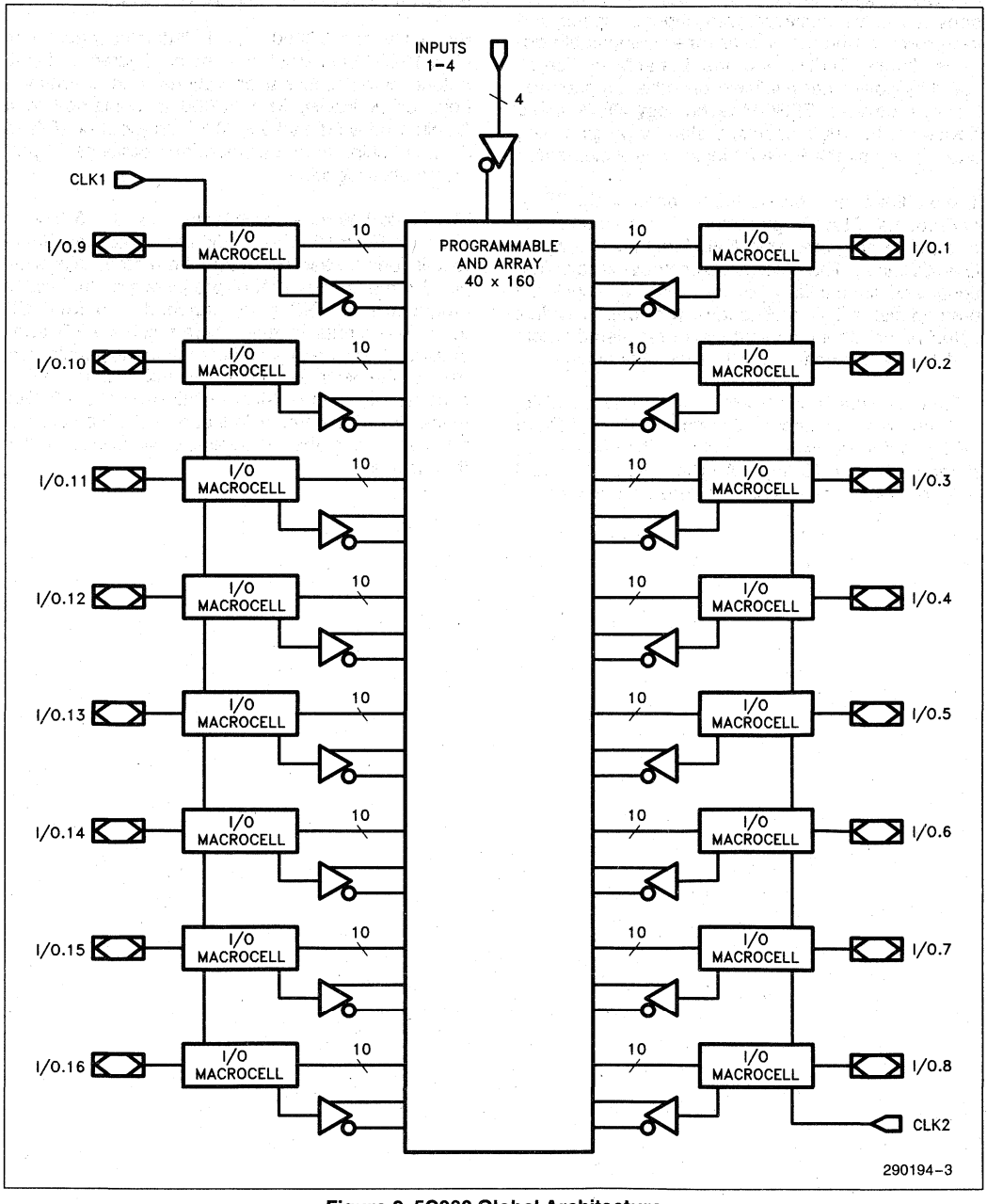


Figure 2. 5C060 Global Architecture

The 5C060 EPLD uses CHMOS EPROM (floating gate) cells as logic control elements instead of fuses. The CHMOS EPROM technology reduces power consumption of EPLDs to less than 20% of a comparable bipolar device without sacrificing speed performance. In addition, Intel's advanced CHMOS II-E EPROM process technology enables greater logic densities to be achieved with superior speed and low-power performance over other comparable devices. Intel's ELPDs add the benefits of "zero" stand-by power not available on other programmable logic devices. EPROM technology allows these devices to be 100% factory tested by programming and erasing all the EPROM logic control elements.

The architecture of the 5C060 is based on the "Sum of Products" PLA (Programmable Logic Array) structure with a programmable AND array feeding into a fixed OR array. The device accommodates combinational and sequential logic functions. A proprietary programmable I/O architecture provides individual selection of either combinatorial or registered output and feedback signals all with selectable polarity.

A feature unique to the 5C060 is the ability to individually program the output registers as a D-, T-, SR-, or JK-type Flip-Flop without sacrificing the utilization of programmable AND logic. Additionally, each output register can be individually clocked from any of the input or feedback paths available within the AND array. With these features, a wide variety of logic functions can be simultaneously implemented–all on the same device.

ARCHITECTURE DESCRIPTION

Externally, the 5C060 has 4 dedicated data input pins, 16 I/O pins which may be configured for input, output, or bidirectional operations, and 2 synchronous clock inputs. The 5C060 is contained in a 24-pin windowed package (0.3 inch wide) or 28-lead J-leaded chip carrier package, and contains 16 programmable registers.

The basic Macrocell architecture for the 5C060 is shown in Figure 3. The 5C060 has 16 of these Macrocells (one for each I/O pin). The Macrocell is organized in the familiar sum-of-products structure with a programmable AND array attached to a fixed OR term. The inputs to the programmable AND array originate from the true and complement signals from each of the dedicated input pins and each of the I/O control blocks. The 40-input AND array of the 5C060 feeds 160 AND gates (product terms) which are distributed among the 16 available Macrocells within that device.

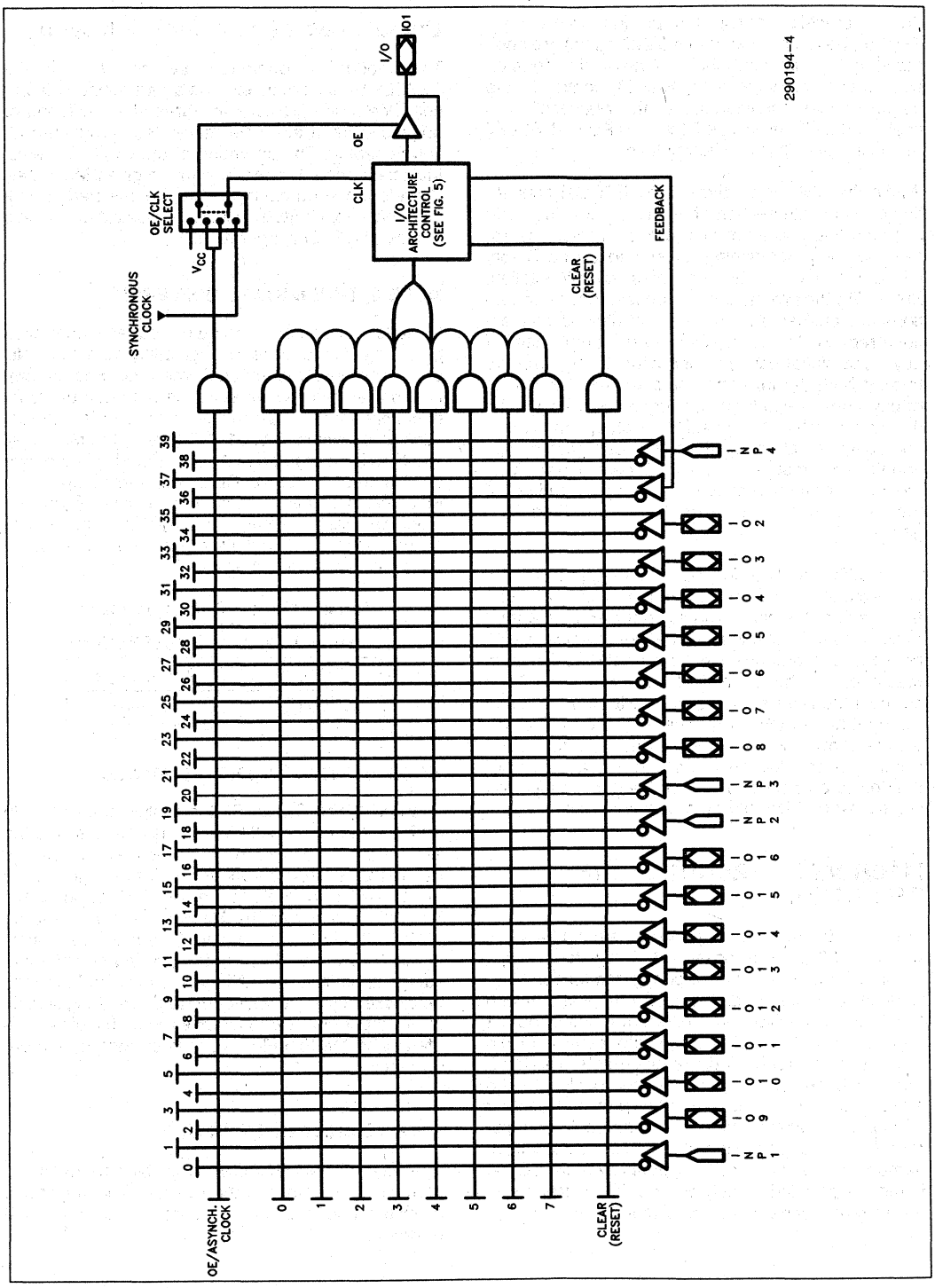


Figure 3. 5C060 Macrocell Architecture

The Macrocells contain ten product terms total. Eight of the ten product terms (AND gates) are dedicated for logic implementation. One product term on each Macrocell is used for RESET control to the output register associated with the Macrocell. The final product term is used for OUTPUT ENABLE/ Asynchronous Clock implementation.

Within the AND array, there is an EPROM connection at every intersection of an input signal (true and complement) and a product term to a given Macrocell. Before programming an erased device, every EPROM connection is made at every intersection. But during the programming process, these connections are opened so that only the desired connections remain. Therefore, the true or complement of any input signal can be connected to any product term. If both the true and complement connections of any signal are left intact, a logical false results on the output of the AND gate. However, if both the true and complement connections are open, then a logic "don't care" results on the AND gate. Lastly, if all the inputs of a product term are programmed open, then a logical true results on the output of the AND gate.

The 5C060 has two dedicated clock inputs to provide synchronous clock signals to the internal registers. Each of the clock signals controls half the total registers within the given device. For example, CLK1 provides synchronous clocking to the registers in Macrocells in the left half of the array while CLK2 controls the registers associated with Macrocells in the right half of the array. The advanced I/O architecture allows for any number of the registers to be synchronously clocked (from none to all). Both of the dedicated clock inputs latch the data into a given register when triggered on a positive edge.

MACROCELL ARCHITECTURE SELECTION

The 5C060 architecture provides each Macrocell with over 50 different possible I/O register configurations. Each I/O pin can be configured for combinatorial or registered output (true or complement) with feedback. In addition, four different types of output registers can be implemented into every I/O pin without any additional logic requirements. The feedback mechanism for each register back into the AND array can be programmed to provide for either registered feedback from the Macrocell or input feedback (treating the pin as an input). Another advantage of the advanced I/O capability of the 5C060 is the ability to individually clock each internal register from asynchronous clock signals.

Output Enable (OE)/Clock Selection

Two modes of operation are provided by the OE/CLK Select Multiplexer as a part of each Macrocell. One mode provides for three-state buffering of outputs while in the other mode, the outputs are always enabled. The operation of the OE/CLK Select Multiplexer sets the mode within a given Macrocell. Therefore, the output mode can be selected individually on every output. Figure 4 illustrates the two modes of OE/CLK operation.

MODE 0: THREE-STATE BUFFERING

In Mode 0, the three-state output buffer is controlled by a single product term originating from the AND array. The output is enabled when the product term is a logical true. Conversely, the output appears as high impedance when the product term is a logical false as shown in Table 1. In Mode 0, the Macrocell Flip-Flop is connected to its associated synchronous clock (either CLK1 or CLK2 depending upon the Macrocell's location within the device). Thus, the Macrocell Flip-Flop may be clocked by its respective synchronous clock but its output will not become valid until the output is enabled.

Table 1. Mode 0 Output Selection

Product Term	Output Buffer
FALSE	Three-State
TRUE	Enabled

MODE 1: OUTPUT BUFFER ENABLED

In Mode 1, the Output Buffer is always enabled. In addition, the Macrocell Flip-Flop is connected to the AND array. The Macrocell Flip-Flop may now be triggered from an asynchronous clock signal generated by the AND array logic to the OE/CLK multiplexable term. Mode 1 allows the Macrocell Flip-Flops to be individually clocked from any of the available signals in the AND array. Since both true and complement values appear in the AND array, the Flip-Flop may be clocked by any positive- or negative-going signals at any input pin. Gated clock structures can be created since the Flip-Flop clock is created by a product term.

Invert Select EPROM Bit

The Invert Select EPROM bit is used to invert the product term input into the register. This applies to all inputs including double inputs on the JK and SR registers.

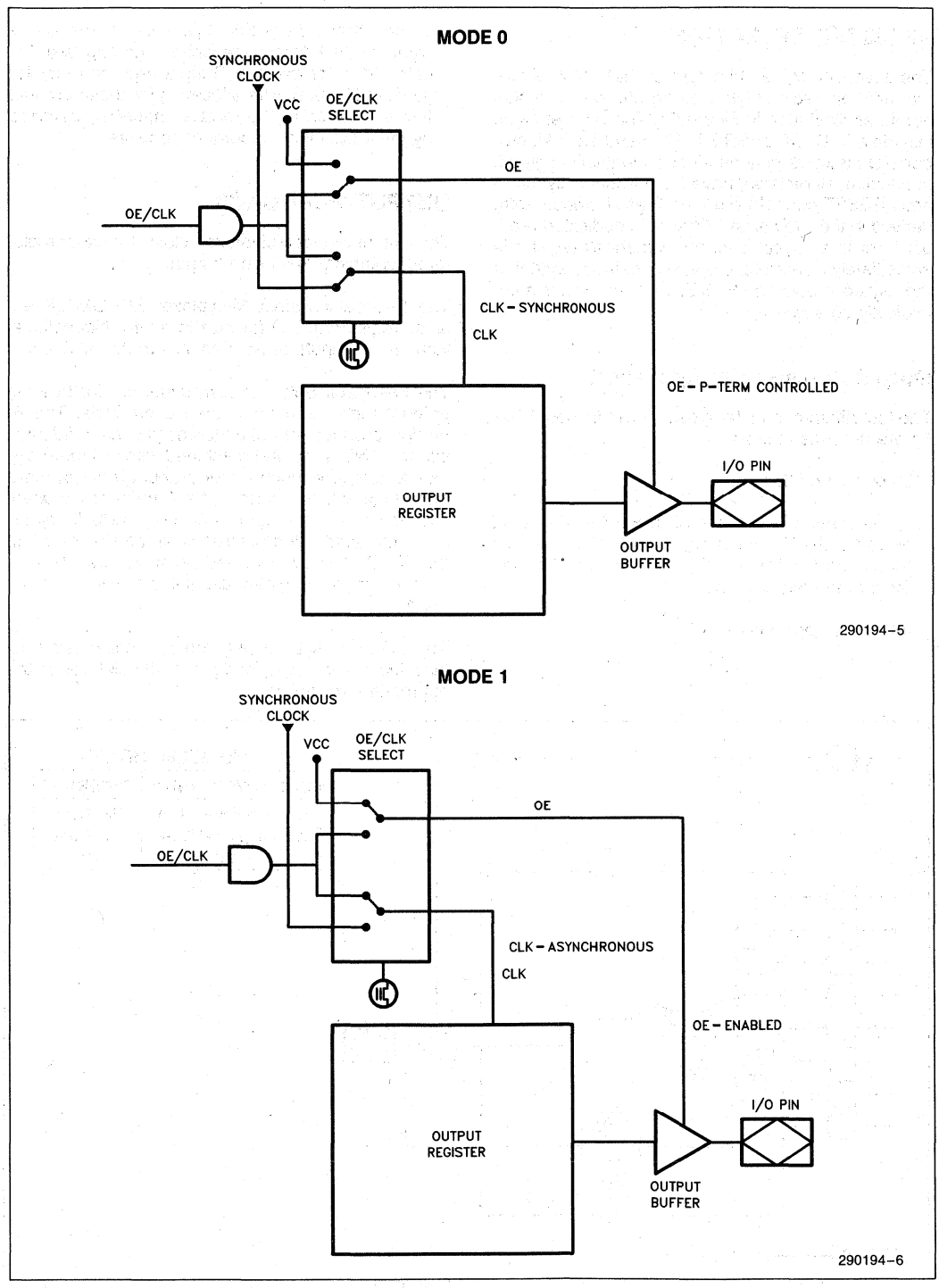


Figure 4. Output Enable/Clock Configuration

REGISTER SELECTION

The advanced I/O architecture of the 5C060 allows four different register types along with combinatorial output as illustrated in Figure 5a. The register types include a T, D, JK, or SR Flip-Flop and each Macrocell I/O structure may be independently configured. In addition, all registers have an individual asynchronous RESET control from a dedicated product term derived in the AND array. When this dedicated product term is a logical one, the Macrocell register is immediately cleared to a logical zero independent of the register clock. The RESET function occurs automatically on power-up.

Output Register Configuration

The four different register types shown in Figure 5b–5e are described below.

D- or T-type Flip-Flops

When either a D- or T-type Flip-Flop is configured as part of the I/O structure, all eight of the product terms into the Macrocell are ORed together and fed into the register input.

JK or SR Registers

When either a JK or SR register is configured, the eight product terms are shared among two OR gates (one for the J or S input and the other for the K or R input). The allocation for these product terms for each of the register inputs is optimized by the iPLDS II development software.

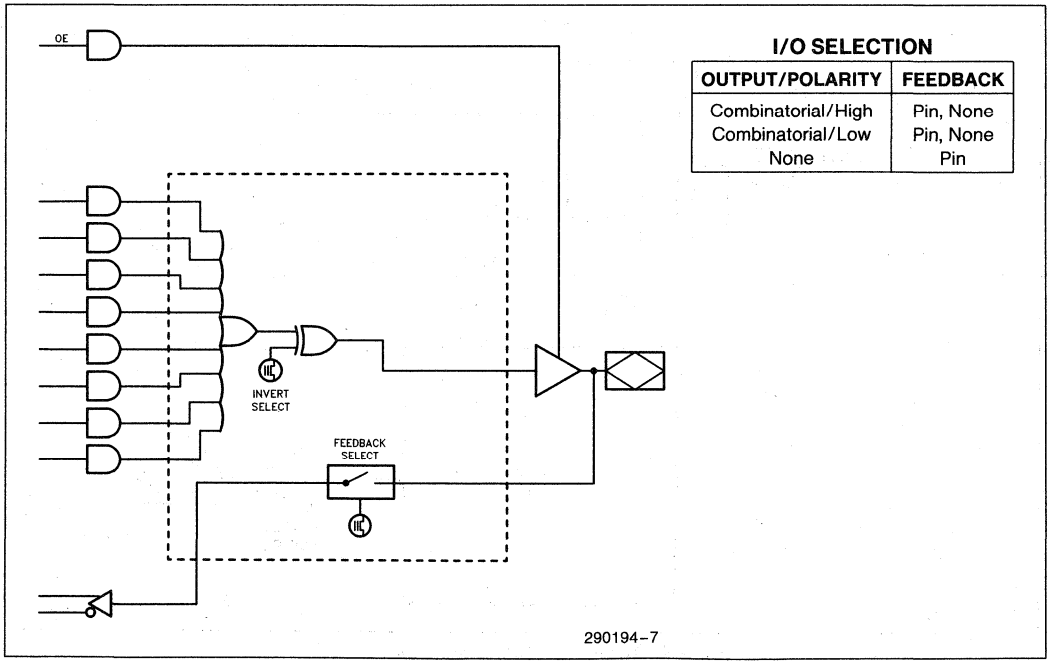
OUTPUT/FEEDBACK

The Output Select Multiplexer allows for either registered, combinatorial or no output.

The Feedback Select Multiplexer EPROM bit enables registered, I/O (using the pin for bidirectional input or just input), or no feedback to the AND array.

The Feedback Select is also important for building product terms with more than 8 products. The 8-product product term of a Macrocell can be fed back into the AND array and combined with still more signals to create a much larger product term (of more than 8-inputs). In addition, if the feedback product term is not to be output, then the iPLDS II will reserve the associated Macrocell pin and indicate it in the REPORT file. A reserved pin should be left floating (no connect) when assembled onto a circuit board.

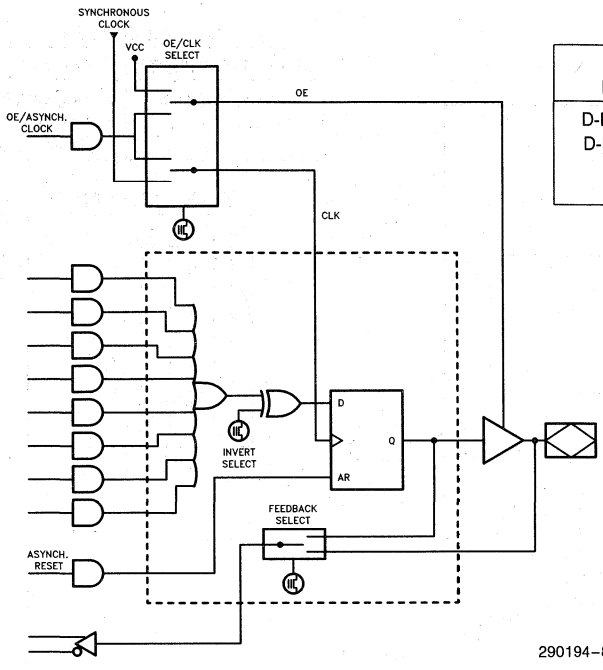
Any I/O pin may be configured as a dedicated input by selecting no output and pin feedback through the appropriate multiplexers.



I/O SELECTION

OUTPUT/POLARITY	FEEDBACK
Combinatorial/High	Pin, None
Combinatorial/Low	Pin, None
None	Pin

Figure 5a. Combinatorial I/O Configuration



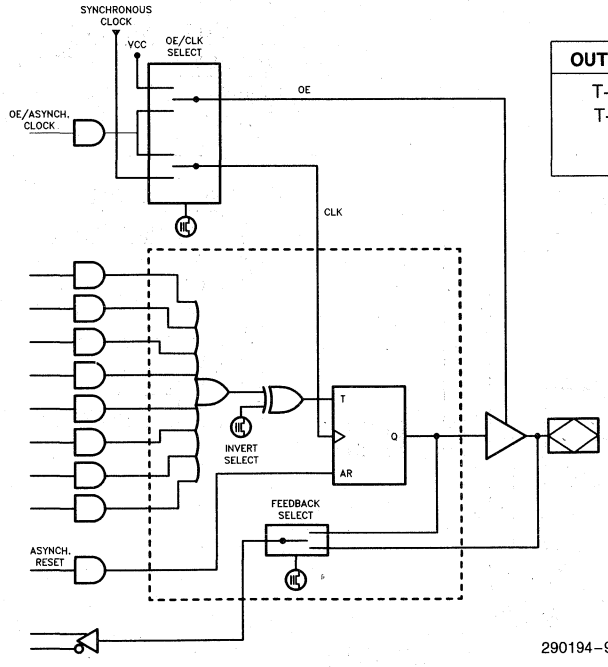
I/O SELECTION

OUTPUT/ POLARITY	FEEDBACK
D-Register/High	D-Register, Pin, None
D-Register/Low	D-Register, Pin, None
None	D-Registered
None	Pin

FUNCTION TABLE

D	Q_n	$Q_n + 1$
0	0	0
0	1	0
1	0	1
1	1	1

Figure 5b. D-Type Flip-Flop Register Configuration



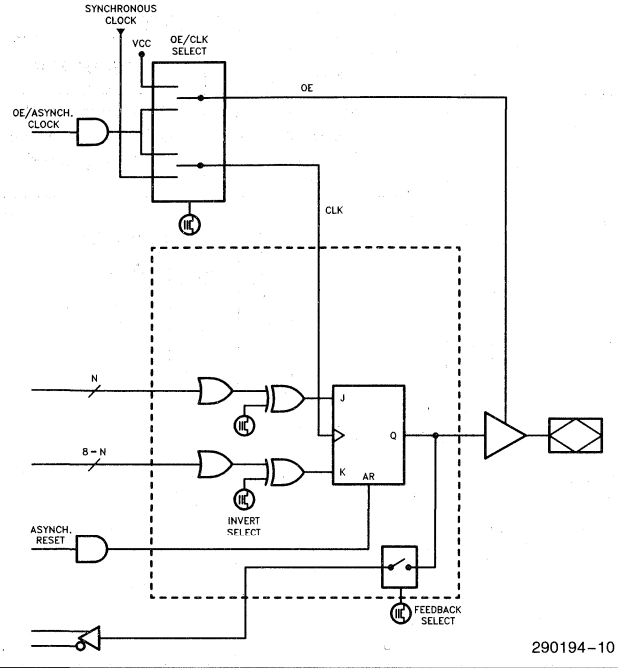
I/O SELECTION

OUTPUT/POLARITY	FEEDBACK
T-Register/High	T-Register, Pin, None
T-Register/Low	T-Register, Pin, None
None	T-Register
None	Pin

FUNCTION TABLE

T	Q_n	$Q_n + 1$
0	0	0
0	1	1
1	0	1
1	1	0

Figure 5c. Toggle Flip-Flop Register Configuration



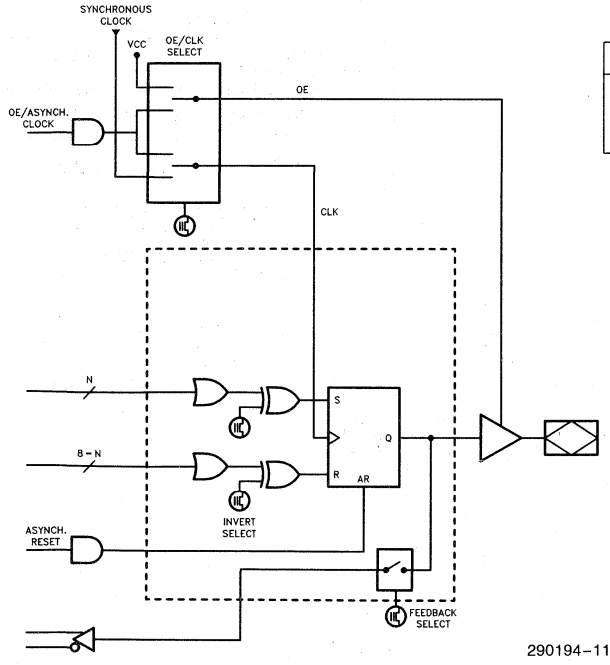
I/O SELECTION

OUTPUT/POLARITY	FEEDBACK
JK Register/High	JK Register, None
JK Register/Low	JK Register, None
None	JK Register

FUNCTION TABLE

J	K	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Figure 5d. JK Flip-Flop Register Configuration



I/O SELECTION

OUTPUT/POLARITY	FEEDBACK
SR Register/High	SR Register, None
SR Register/Low	SR Register, None
None	SR Register

FUNCTION TABLE

S	R	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	Illegal	

Figure 5e. SR Flip-Flop Register Configuration

Erased-State Configuration

Prior to programming or after erasing, the I/O structure is configured for combinatorial active low output with input (pin) feedback.

ERASURE CHARACTERISTICS

Erasure characteristics of the device are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000Å. It should be noted that sunlight and certain types of flourescent lamps have wavelengths in the 3000Å–4000Å. Data shows that constant exposure to room level flourescent lighting could erase the typical device in approximately three years, while it would take approximately one week to cause erasure when exposed to direct sunlight. If the 5C060 is to be exposed to these types of lighting conditions for extended periods of time, conductive opaque labels should be placed over the device window to prevent unintentional erasure.

The recommended erasure procedure for the 5C060 is exposure to shortwave ultraviolet light with a wavelength of 2537Å. The integrated dose (i.e., UV intensity × exposure time) for erasure should be a minimum of fifteen (15) Wsec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12,000 μW/cm^2 power rating. The 5C060 should be placed within one inch of the lamp tubes during erasure. The maximum integrated dose the 5C060 can be exposed to without damage is 7258 Wsec/cm^2 (1 week at 12,000 μW/cm^2). Exposure to high intensity UV light for longer periods may cause permanent damage to the device.

PROGRAMMING CHARACTERISTICS

Initially, and after erasure, all the EPROM control bits of the 5C060 are connected (in the "1" state). Each of the connected control bits are selectively disconnected by programming the EPROM cells into their "0" state. Programming voltage and waveform specifications are available by request from Intel to support programming of the 5C060.

int$_e$ligent Programming™ Algorithm

The 5C060 supports the int$_e$ligent Programming Algorithm which rapidly programs Intel ELPDs using an efficient and reliable method. The int$_e$ligent Programming Algorithm is particularly suited to the production programming environment. This method ensures reliability as the incremental program margin of each bit is continually monitored to determine when the bit has been successfully programmed.

FUNCTIONAL TESTING

Since the logical operation of the 5C060 is controlled by EPROM elements, the device is completely testable. Each programmable EPROM bit controlling the internal logic is tested using application-independent test program patterns. After testing, the devices are erased before shipment to customers. No post-programming tests of the EPROM array are required.

The testability and reliability of EPROM-based programmable logic devices is an important feature over similar devices based on fuse technology. Fuse-based programmable logic devices require a user to perform post-programming tests to insure proper programming. These tests must be done at the device level because of the cummulative error effect. For example, a board containing ten devices each possessing a 2% device fallout translates into an 18% fallout at the board level (it should be noted that programming fallout of fuse-based programmable logic devices is typically 2% or higher).

DESIGN RECOMMENDATIONS

For proper operation, it is recommended that all input and output pins be constrained to the voltage range GND < (V_{IN} or V_{OUT}) < V_{CC}. Unused inputs and I/Os should be tied to V_{CC} or GND to minimize device power consumption. Reserved pins (as indicated in the logic compiler REPORT file) should be left floating (no connect) so that the pin can attain the appropriate logic level. A power supply decoupling capacitor of at least 0.2 μF must be connected directly between V_{CC} and GND pins of the device.

As with all CMOS devices, ESD handling procedures should be used with the 5C060 to prevent damage to the device during programming, assembly, and test.

DESIGN SECURITY

A single EPROM bit provides a programmable design security feature that controls the access to the data programmed into the device. If this bit is set, a proprietary design within the device cannot be copied. This EPROM security bit enables a higher degree of design security than fused-based devices

since programmed data within EPROM cells is invisible even to microscopic evaluation. The EPROM security bit, along with all the other EPROM control bits, will be reset by erasing the device.

AUTOMATIC STAND-BY MODE

The 5C060 contains a programmable bit, the Turbo Bit, that optimizes operation for speed or for power savings. When the Turbo Bit is programmed (TURBO = ON), the device is optimized for maximum speed. When the Turbo Bit is not programmed (TURBO = OFF), the device is optimized for power savings by entering standby mode during periods of inactivity.

Figure 6 shows the device entering standby mode approximately 100 ns after the last input transition. When the next input transition is detected, the device returns to active mode. Wakeup time adds an additional 25 ns to the propagation delay through the device as measured from the first input. No delay will occur if an output is dependent on more than one input and the last of the inputs changes after the device has returned to active mode.

After erasure, the Turbo Bit is unprogrammed (OFF); automatic standby mode is enabled. When the Turbo Bit is programmed (ON), the device never enters standby mode.

LATCH-UP IMMUNITY

All of the input, I/O, and clock pins of the 5C060 have been designed to resist latch-up which is inherent in inferior CMOS structures. The 5C060 is designed with Intel's proprietary CHMOS II-E EPROM process. Thus, each of the pins will not experience latch-up with currents up to ±100 mA and voltages ranging from −1V to (V_{CC} + 1V). Furthermore, the programming pin is designed to resist latch-up to the 13.5V maximum device limit.

INTEL PROGRAMMABLE LOGIC DEVELOPMENT SYSTEM II (iPLDS II)

iPLDS II provides all the tools needed to design with Intel EPLDs or compatible devices. In addition to providing development assistance, iPLDS II insulates the user from having to know all the intricate details of EPLD architecture (the machine will optimize a design to benefit from architectual features). It contains comprehensive third generation software that supports four different design entry methods, minimizes logic, does automatic pin assignments and produces the best design fit for the selected EPLD. It is user friendly with guided menus, on-line Help messages and soft key inputs.

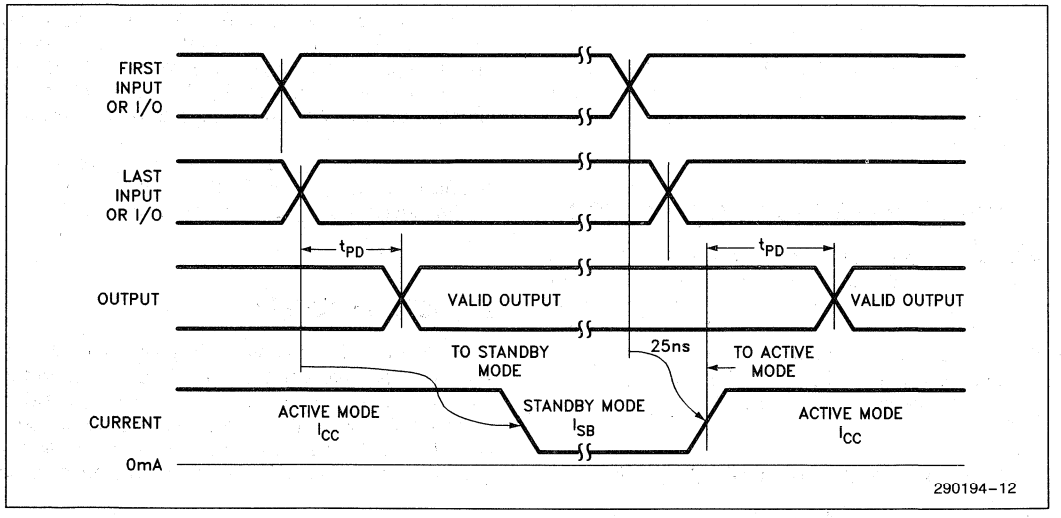


Figure 6. 5C060 Standby and Active Mode Transitions

In addition, the iPLDS II contains programmer hardware in the form of an iUP-PC Universal Programmer Personal Computer to enable the user to program EPLDs, read and verify programmed devices and also to graphically edit programming files. The software generates industry standard JEDEC object code output files which can be downloaded to other programmers as well.

iPLS II software interfaces to several schematic capture packages to enable designs to be entered in schematic form. IPLDview-286/IPLDdraw allows the designer to use familiar TTL symbols or EPLD design primitive symbols. User-defined symbols are also supported. IPLD draw provides a path to A.C. timing simulation of EPLD designs.

SCHEMA III-PLD allows the designer to use TTL symbols, EPLD custom macros, or EPLD design primitive symbols. It also supports user-defined symbols.

Other design formats include Boolean equation entry (supported directly by iPLS II) and state machine entry (supported by iSTATE).

Detailed information on the Intel Programmable Logic Development System II is contained in a separate Intel data sheet. (Order Number: 280168). Refer to the tools section of the *Programmable Logic* handbook for a complete listing of development tools.

The 5C060 is also supported by third-party logic compilers such as ABEL*, CUPL*, PLDesigner*, Log/IC, etc. Programming support is provided by third-party programmer companies such as Data I/O, Logical Devices, STAG, etc. Please refer to the "Third-Party Support" lists in the *Programmable Logic* handbook for complete information and vendor contacts.

ADF PRIMITIVES SUPPORTED

The following ADF primitives are supported by this device:

INP	JOJF
CONF	JONF
COIF	SONF
RONF	SOSF
RORF	TOIF
ROIF	TONF
NORF	TOTF
NOJF	CLKB
NOSF	
NOTF	

ORDERING INFORMATION

t_{PD} (ns)	t_{CO} (ns)	f_{MAX} (MHz)	Order Code	Package	Operating Range
45	22	26	D5C060-45	CERDIP	Commercial
			P5C060-45	PDIP	
			N5C060-45	PLCC	
55	25	23	D5C060-55	CERDIP	Commercial
			P5C060-55	PDIP	
			N5C060-55	PLCC	

*Abel is a trademark of Data I/O, Corp. CUPL is a trademark of Logical Devices, Inc. PLDesigner is a trademark of MINC, Inc.

ABSOLUTE MAXIMUM RATINGS*

Symbol	Parameter	Min	Max	Units
V_{CC}	Supply Voltage(1)	−2.0	7.0	V
V_{PP}	Programming Supply Voltage(1)	−2.0	13.5	V
V_I	DC Input Voltage(1)(2)	−0.5	V_{CC}+0.5	V
t_{stg}	Storage Temperature	−65	+150	°C
t_{amb}	Ambient Temperature(3)	−10	+85	°C

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

NOTES:
1. Voltages with respect to ground.
2. Minimum DC input is −0.5V. During transitions, the inputs may undershoot to −2.0V or overshoot to 7.0V for periods less than 20 ns under no load conditions.
3. Under bias. Extended temperature versions are also available.

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Min	Max	Unit
V_{CC}	Supply Voltage	4.75	5.25	V
V_{IN}	Input Voltage	0	V_{CC}	V
V_O	Output Voltage	0	V_{CC}	V
T_A	Operating Temperature	0	+70	°C
t_R(4)	Input Rise Time		500	ns
t_F(4)	Input Fall Time		500	ns

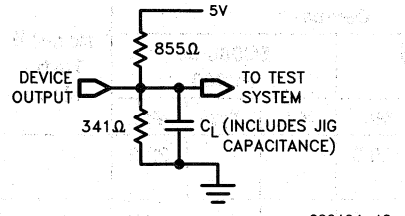
NOTE:
4. t_R, t_F for CLK is 250 ns max.

D.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5.0V ±5%

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
V_{IH}(5)	HIGH Level Input Voltage			2.0		V_{CC} + 0.3	V
V_{IL}(5)	LOW Level Input Voltage			−0.3		0.8	V
V_{OH}(6)	HIGH Level Output Voltage	I_O = −4.0 mA DC, V_{CC} = Min.		2.4			V
V_{OL}	LOW Level Output Voltage	I_O = 4.0 mA DC, V_{CC} = Min.				0.45	V
I_I	Input Leakage Current	V_{CC} = Max., GND < V_{IN} < V_{CC}				±10.0	µA
I_{OZ}	Output Leakage Current	V_{CC} = Max., GND < V_{OUT} < V_{CC}				±10.0	µA
I_{SC}(7)	Output Short Circuit Current	V_{CC} = Max., V_{OUT} = 0.5V			20	30	mA
I_{SB}(8)	Standby Current (Standby)	V_{CC} = Max., V_{IN} = V_{CC} or GND			50	100	µA
I_{CC}	Power Supply Current (Active) (Turbo Bit Off) Device Prog. as 16-Bit Ctr. (See I_{CC} vs. Freq. Graph.)	V_{CC} = Max., V_{IN} = V_{CC} or GND	No Load, Input Freq. = 1 MHz		10	15	mA

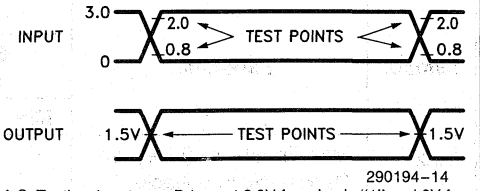
NOTES:
5. Absolute values with respect to device GND; all over and undershoots due to system or tester noise are included.
6. I_O at CMOS levels (3.84V) = −2 mA.
7. Not more than 1 output should be tested at a time. Duration of that test must not exceed 1 second.
8. With Turbo Bit Off, device automatically enters standby mode approximately 100 ns after last input transition.

A.C. TESTING LOAD CIRCUIT



C_L = 50 pF

A.C. TESTING INPUT, OUTPUT WAVEFORM



A.C. Testing: Inputs are Driven at 3.0V for a Logic "1" and 0V for a Logic "0". Timing Measurements are made at 2.0V for a Logic "1" and 0.8V for a Logic "0" on inputs. Outputs are measured at a 1.5V point. Device input rise and fall times < 6 ns.

CAPACITANCE

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
C_{IN}	Input Capacitance	V_{IN} = 0V, f = 1.0 MHz			20	pF
C_{OUT}	Output Capacitance	V_{OUT} = 0V, f = 1.0 MHz			20	pF
C_{CLK}	Clock Pin Capacitance	V_{IN} = 0V, f = 1.0 MHz			20	pF
C_{VPP}	V_{PP} Pin	CLK2 on 5C060, f = 1.0 MHz			50	pF

A.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ±5%, Turbo Bit On[9]

Symbol	From	To	Device						Non-[11] Turbo Mode	Unit
			5C060-45 EP600-3			5C060-55 EP600				
			Min	Typ	Max	Min	Typ	Max		
t_{PD1}	Input	Comb. Output			43			53	+25	ns
t_{PD2}	I/O	Comb. Output			45			55	+25	ns
t_{PZX}[10]	I or I/O	Output Enable			45			55	+25	ns
t_{PXZ}[10]	I or I/O	Output Disable			45			55	+25	ns
t_{CLR}	Asynch. Reset	Q Reset			45			55	+25	ns

NOTES:
9. Typical Values are at T_A = 25°C, V_{CC} = 5V, Active Mode.
10. t_{PZX} and t_{PXZ} are measured at ±0.5V from steady state voltage as driven by spec. output load. t_{PXZ} is measured with C_L = 5 pF.
11. If device is operated with Turbo Bit Off (Non-Turbo Mode), and the device has been inactive for approx. 100 ns, increase time by amount shown.

SYNCHRONOUS CLOCK MODE A.C. CHARACTERISTIC

T_A = 0°C to 70°C, V_{CC} = 5.0V ±5%, Turbo Bit On[9]

Symbol	Parameter	Device						Non-[11] Turbo Mode	Unit
		5C060-45 EP600-3			5C060-55 EP600				
		Min	Typ	Max	Min	Typ	Max		
f_{MAX}	Max. Frequency (Pipelined) ($1/t_{SU}$—No Feedback)			26.3			23.3		MHz
f_{CNT}	Max. Count Frequency ($1/t_{CNT}$—With Feedback)			22.2			18.2		MHz
t_{SU1}	Input Setup Time to CLK	36			41			+25	ns
t_{SU2}	I/O Setup Time to CLK	38			43			+25	ns
t_H	I or I/O Hold after CLK High	0			0				ns
t_{CO}	CLK High to Output Valid			22			25		ns
t_{CNT}	Register Output Feedback to Register Input—Internal Path	45			55			+25	ns
t_{CH}	CLK High Time	17.5			21.5				ns
t_{CL}	CLK Low Time	17.5			21.5				ns

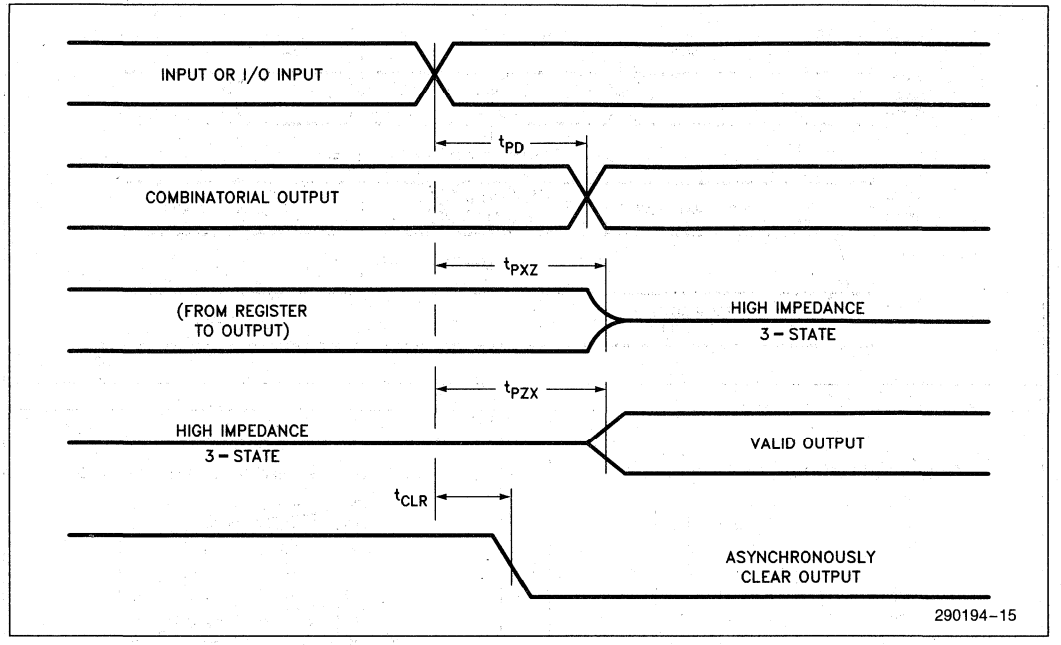
ASYNCHRONOUS CLOCK MODE A.C. CHARACTERISTICS

T_A = 0°C to 70°C, V_{CC} = 5.0V ±5%, Turbo Bit On[8]

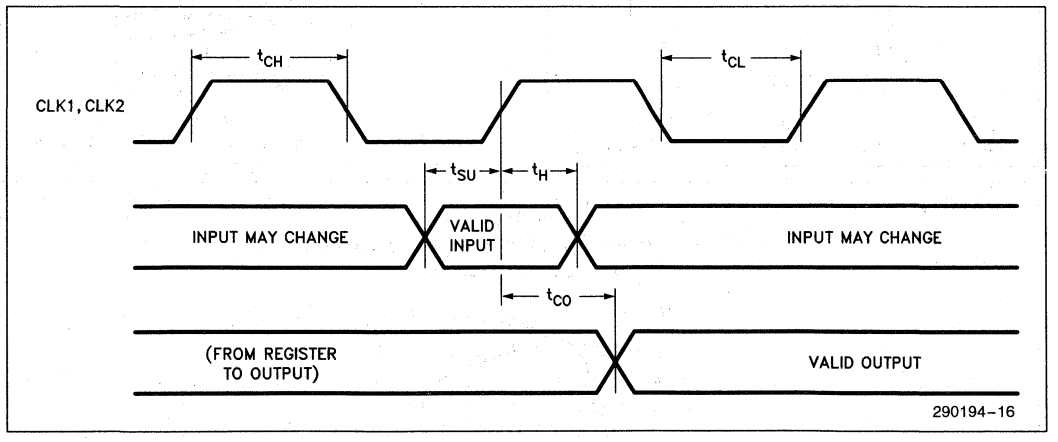
Symbol	Parameter	Device						Non-[11] Turbo Mode	Unit
		5C060-45 EP600-3			5C060-55 EP600				
		Min	Typ	Max	Min	Typ	Max		
f_{ACNT}	Max. Count Frequency ($1/t_{ACNT}$—With Feedback)			22.2			18.2		MHz
t_{ASU1}	Input Setup Time to Asynch. Clock	10			10			+25	ns
t_{ASU2}	I/O Setup Time to Asynch. Clock	12			12			+25	ns
t_{AH}	Input or I/O Hold After Asynch. Clock	15			15				ns
t_{ACO}	Asynch. CLK to Output Valid			50			58	+25	ns
t_{ACNT}	Register Output Feedback to Register Input—Internal Path	45			55			+25	ns
t_{ACH}	Asynch. CLK High Time	17.5			21.5			+25	ns
t_{ACL}	Asynch. CLK Low Time	17.5			21.5			+25	ns

SWITCHING WAVEFORMS

COMBINATORIAL MODE

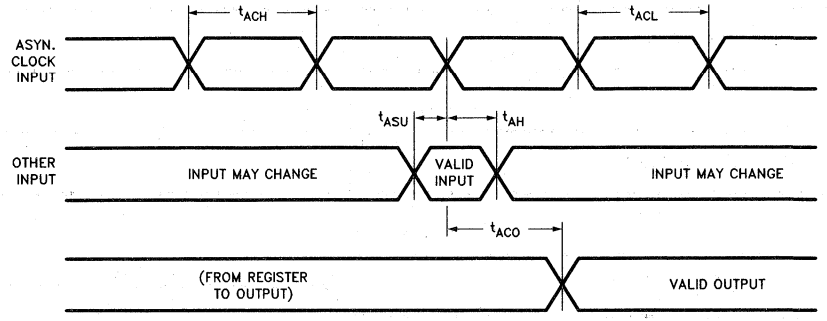


SYNCHRONOUS CLOCK MODE



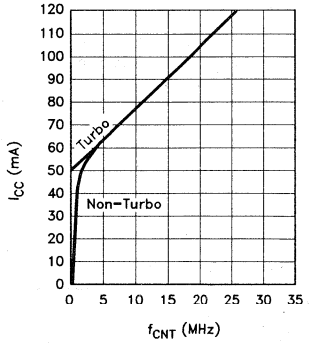
SWITCHING WAVEFORMS (Continued)

ASYNCHRONOUS CLOCK MODE



290194-17

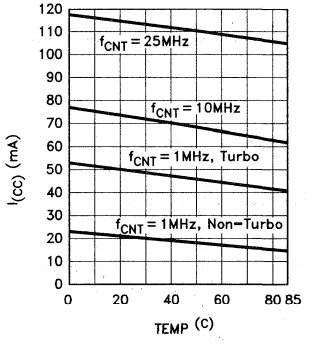
5C060
Current in Relation to Frequency



Conditions: T_A = 0°C, V_{CC} = 5.25V

290194-18

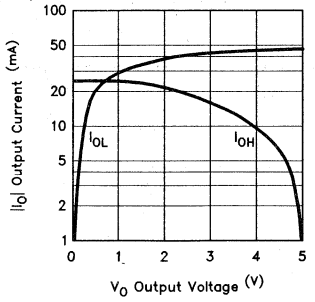
5C060
Current in Relation to Temperature



Conditions: V_{CC} = 5.25V, TTL inputs

290194-19

5C060
Output Drive Current in Relation to Voltage

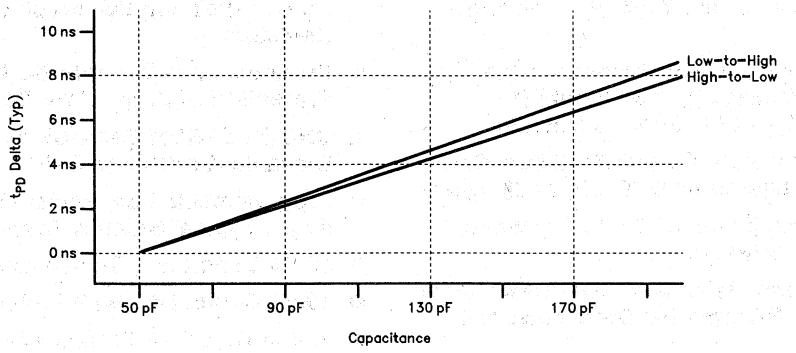


Conditions: T_A = 25°C

290194-20

SWITCHING WAVEFORMS (Continued)

5C060-45
t_{PD} vs Capacitive Loading



290194-21

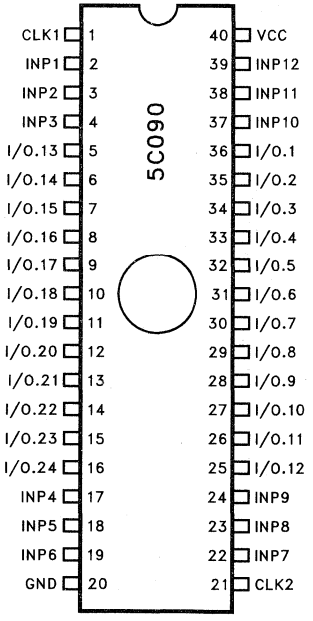
T_A = 25°C
V_{CC} = 5.0V
One output switching; typical path

2

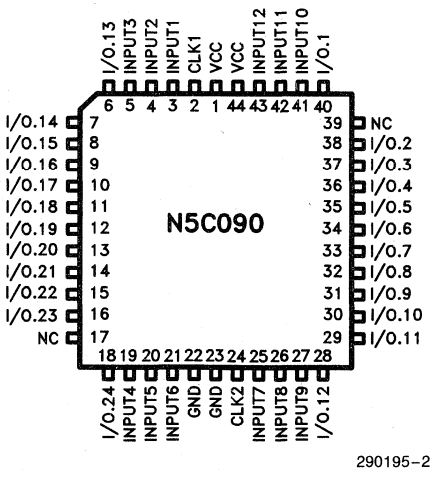
5C090
24-MACROCELL CHMOS EPLD

- High-Performance LSI Semi-Custom Logic Alternative to Low-End Gate Arrays, TTL, and 74HC SSI and MSI Logic
- 24 Macrocells with Programmable I/O Architecture; Up to 36 Inputs (12 Dedicated, 24 I/O) or 24 Outputs
- Programmable Output Registers. Can be Configured as D, T, SR, or JK Types
- t_{PD} (max) 50 ns, 26.3 MHz Pipelined, 20 MHz w/Feedback
- 8 P-Terms, Selectable SOP Invert, Clear and OE P-Terms for Each Macrocell
- Programmable Clock System with 2 Synchronous Clocks and Asynchronous Clocking Option on all Registers
- Programmable Security Bit Allows Total Protection of Proprietary Designs
- CHMOS EPROM Technology Based. UV Erasable (CerDIP) or OTP
- Programmable Low Power Option; 50 μA Typical Standby Current
- 100% Generically Tested Logic Array
- 100% Compatible with EP900
- 40-Pin CerDIP/PDIP and 44-Pin PLCC Packages

(See Packaging Spec., Order Number #231369)



290195-1



290195-2

Figure 1. 5C090 Pin Configurations

October 1990
Order Number: 290195-003

The Intel 5C090 EPLD (Erasable Programmable Logic Device) is a 24-macrocell, 40-pin, general-purpose device. The device can be used to replace low-end gate arrays, multiple programmable logic arrays and LS TTL and 74HC (CMOS) SSI and MSI logic devices. With its revolutionary programmable I/O architecture, the device has advanced functional capabilities beyond that of typical programmable logic. Figure 2 shows the global architecture of the device.

The 5C090 EPLD uses CHMOS EPROM (floating gate) cells as logic control elements instead of fuses. The CHMOS EPROM technology reduces power consumption of EPLDs to less than 20% of a comparable bipolar device without sacrificing speed performance. In addition, Intel's advanced CHMOS II-E EPROM process technology enables greater logic densities to be achieved with superior speed and low-power performance over other comparable devices. Intel's ELPDs add the benefits of "zero" stand-by power not available on other programmable logic devices. EPROM technology allows these devices to be 100% factory tested by programming and erasing all the EPROM logic control elements.

The architecture of the 5C090 is based on the "Sum of Products" PLA (Programmable Logic Array) structure with a programmable AND array feeding into a fixed OR array. The device accommodates combinational and sequential logic functions. A proprietary programmable I/O architecture provides individual selection of either combinatorial or registered output and feedback signals all with selectable polarity.

A feature unique to the 5C090 is the ability to individually program the output registers as a D-, T-, SR-, or JK-type Flip-Flop without sacrificing the utilization of programmable AND logic. Additionally, each output register can be individually clocked from any of the input or feedback paths available within the AND array. With these features, a wide variety of logic functions can be simultaneously implemented—all on the same device.

ARCHITECTURE DESCRIPTION

The 5C090 has 12 dedicated inputs, 24 I/O pins which may be configured for input, output, or bidirectional operations, and 2 synchronous clock inputs. The 5C090 is packaged in a 40-lead windowed ceramic DIP or 44-lead plastic leaded chip carrier package and contains 24 programmable registers.

The basic Macrocell architecture for the 5C090 is shown in Figure 3. The 5C090 has 24 of these macrocells (one for each I/O pin). The Macrocell is organized in the familiar sum-of-products structure with a programmable AND array attached to a fixed OR term. The inputs to the programmable AND array originate from the true and complement signals from each of the dedicated input pins and each of the I/O control blocks.

The AND array for the 5C090 has 72 inputs derived from the true and complement signals at the input and I/O pins. The AND array in the 5C090 encompasses 240 product terms which are distributed among the 24 Macrocells.

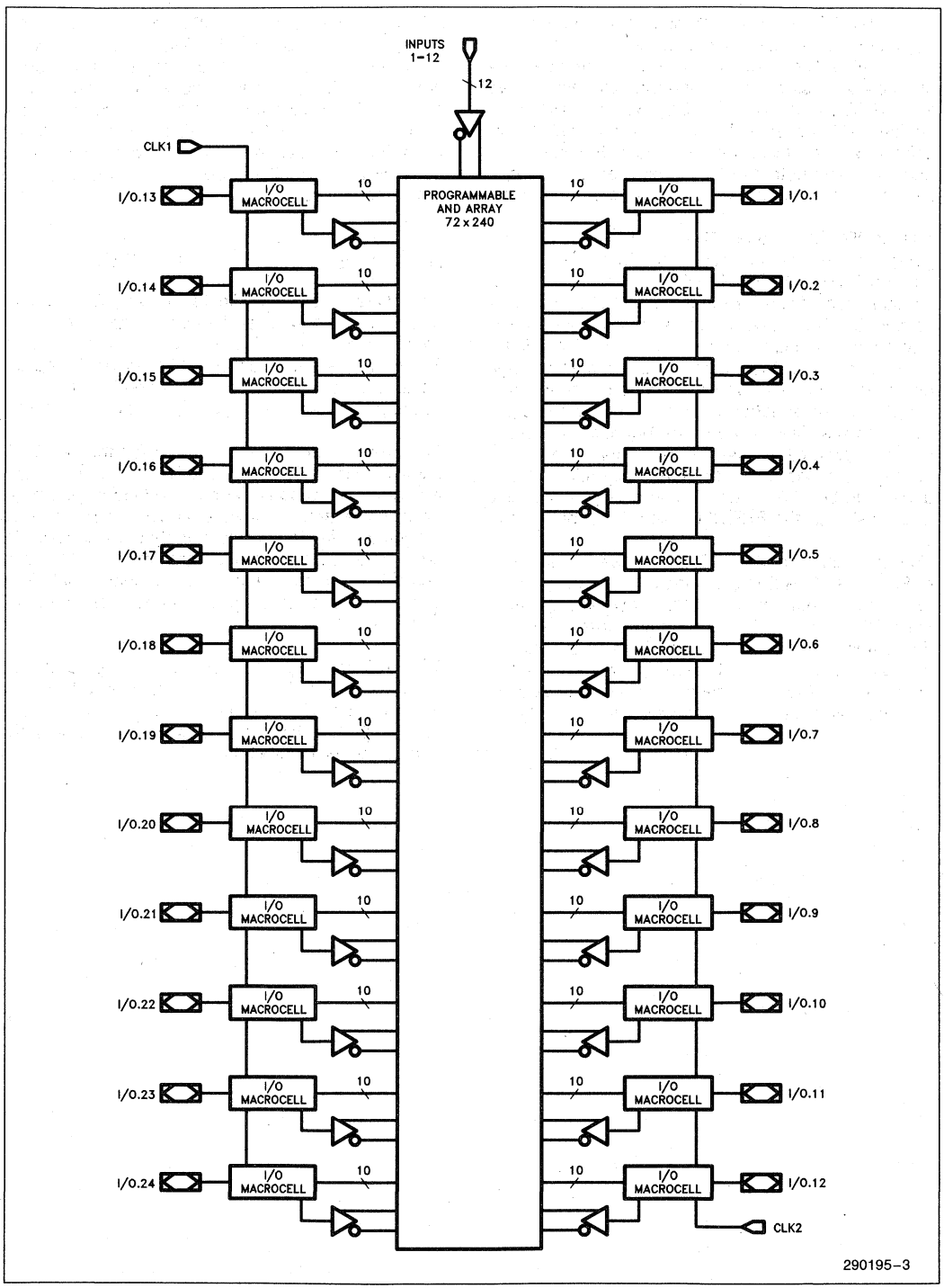


Figure 2. 5C090 Global Architecture

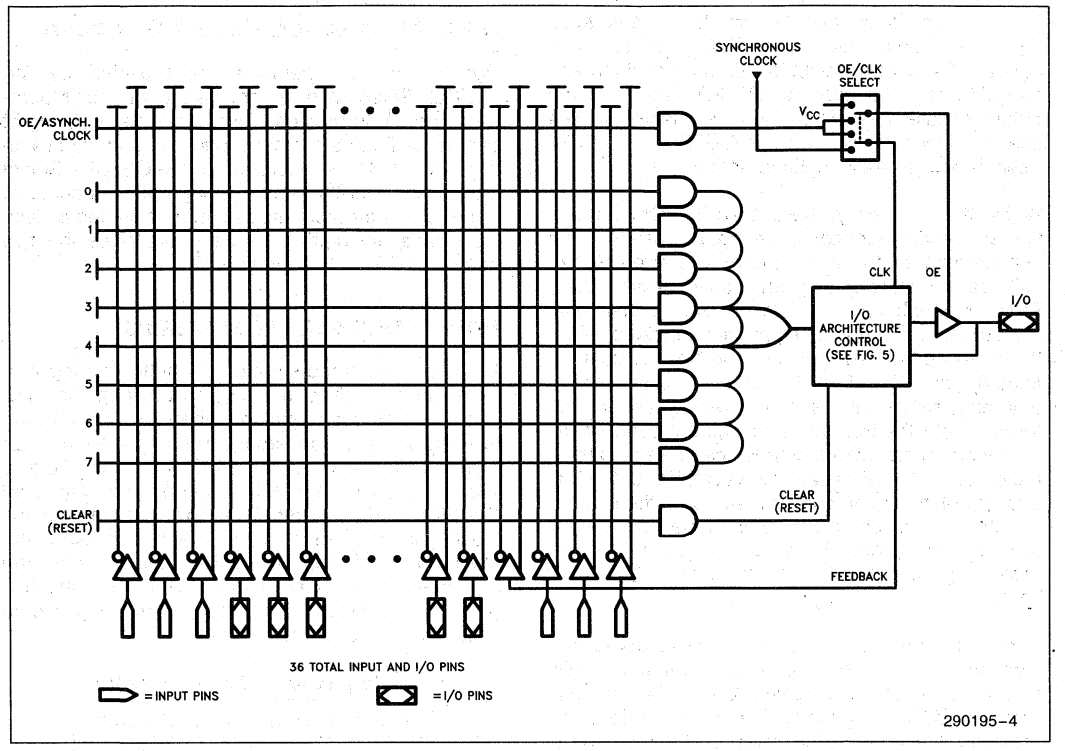


Figure 3. 5C090 Macrocell Architecture

The Macrocells contain ten product terms total. Eight of the ten product terms (AND gates) are dedicated for logic implementation. One product term on each Macrocell is used for RESET control to the output register associated with the Macrocell. The final product term is used for OUTPUT ENABLE/ Asynchronous Clock implementation.

Within the AND array, there is an EPROM connection at every intersection of an input signal (true and complement) and a product term to a given Macrocell. Before programming an erased device, every EPROM connection is made at every intersection. But during the programming process, these connections are opened so that only the desired connections remain. Therefore, the true or complement of any input signal can be connected to any product term. If both the true and complement connections of any signal are left intact, a logical false results on the output of the AND gate. However, if both the true and complement connections are open, then a logic "don't care" results on the AND gate. Lastly, if all the inputs of a product term are programmed open, then a logical true results on the output of the AND gate.

The 5C090 has two dedicated clock inputs to provide synchronous clock signals to the internal registers. Each of the clock signals controls half the total registers within the given device. For example, CLK1 provides synchronous clocking to the registers in Macrocells in the left half of the array while CLK2 controls the registers associated with Macrocells in the right half of the array. The advanced I/O architecture allows for any number of the registers to be synchronously clocked (from none to all). Both of the dedicated clock inputs latch the data into a given register when triggered on a positive edge.

MACROCELL ARCHITECTURE SELECTION

The 5C090 architecture provides each Macrocell with over 50 different possible I/O register configurations. Each I/O pin can be configured for combinatorial or registered output (true or complement) with feedback. In addition, four different types of output registers can be implemented into every I/O pin without any additional logic requirements. The feedback mechanism for each register back into the AND array can be programmed to provide for either registered feedback from the Macrocell or input feedback (treating the pin as an input). Another advantage of the advanced I/O capability of the 5C090 is the ability to individually clock each internal register from asynchronous clock signals.

Output Enable (OE)/Clock Selection

Two modes of operation are provided by the OE/CLK Select Multiplexer as a part of each Macrocell. One mode provides for three-state buffering of outputs while in the other mode, the outputs are always enabled. The operation of the OE/CLK Select Multiplexer sets the mode within a given Macrocell. Therefore, the output mode can be selected individually on every output. Figure 4 illustrates the two modes of OE/CLK operation.

MODE 0: THREE-STATE BUFFERING

In Mode 0, the three-state output buffer is controlled by a single product term originating from the AND array. The output is enabled when the product term is a logical true. Conversely, the output appears as high impedance when the product term is a logical false as shown in Table 1. In Mode 0, the Macrocell Flip-Flop is connected to its associated synchronous clock (either CLK1 or CLK2 depending upon the Macrocell's location within the device). Thus, the Macrocell Flip-Flop may be clocked by its respective synchronous clock but its output will not become valid until the output is enabled.

Table 1. Mode 0 Output Selection

Product Term	Output Buffer
FALSE	Three-State
TRUE	Enabled

MODE 1: OUTPUT BUFFER ENABLED

In Mode 1, the Output Buffer is always enabled. In addition, the Macrocell Flip-Flop is connected to the AND array. The Macrocell Flip-Flop may now be triggered from an asynchronous clock signal generated by the AND array logic to the OE/CLK multiplexable term. Mode 1 allows the Macrocell Flip-Flops to be individually clocked from any of the available signals in the AND array. Since both true and complement values appear in the AND array, the Flip-Flop may be clocked by positive-or negative-going signals at any input pin. Gated clock structures can be created since the Flip-Flop clock is created by a product term.

Invert Select EPROM Bit

The Invert Select EPROM bit is used to invert the product term input into the register. This applies to all inputs including double inputs on the JK and SR registers.

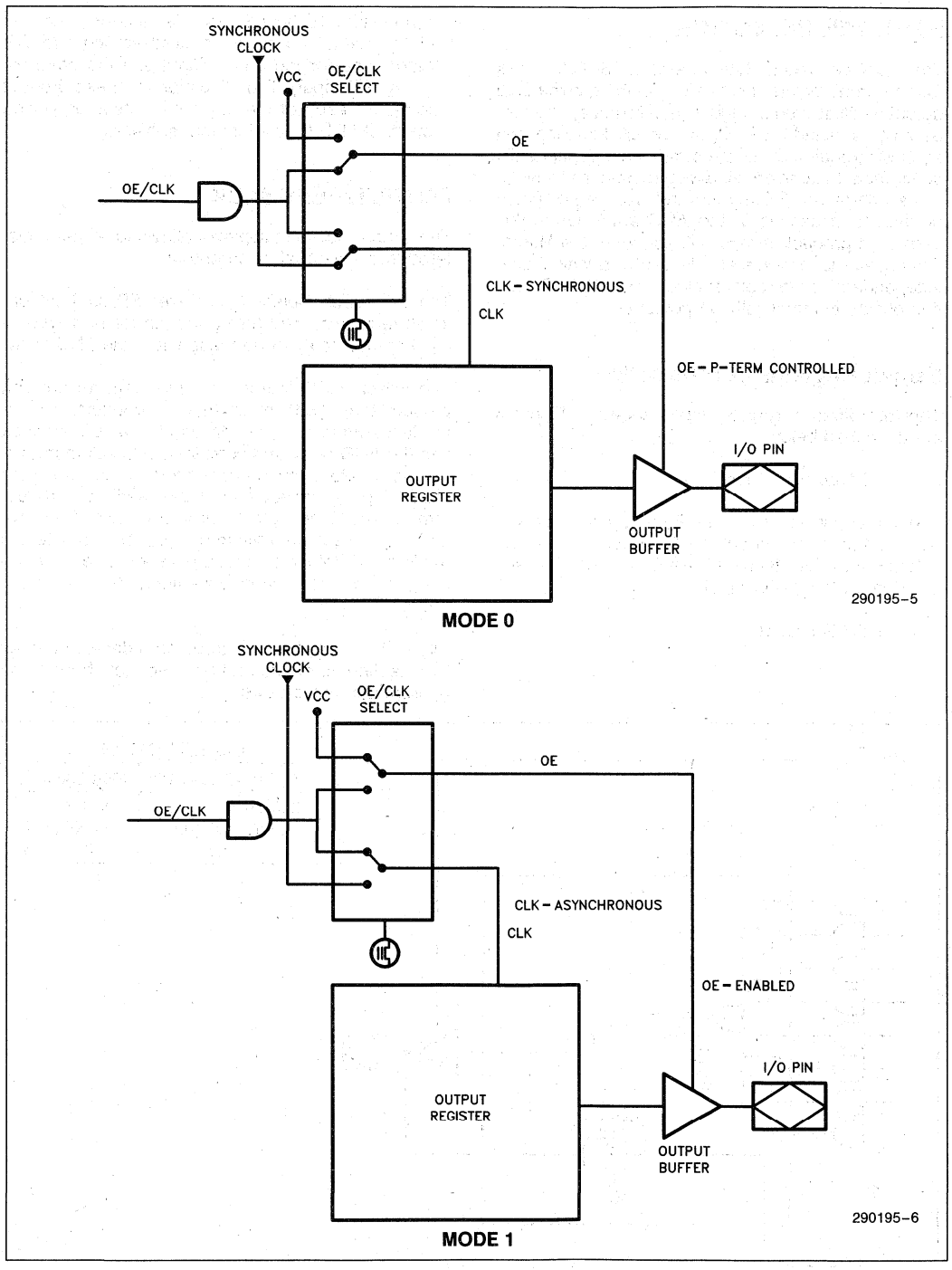


Figure 4. Output Enable/Clock Configuration

REGISTER SELECTION

The advanced I/O architecture of the 5C090 allows four different register types along with combinatorial output as illustrated in Figure 5a through e. The register types include a T, D, JK, or SR Flip-Flop and each Macrocell I/O structure may be independently configured. In addition, all registers have an individual asynchronous RESET control from a dedicated product term derived in the AND array. When this dedicated product term is a logical one, the Macrocell register is immediately cleared to a logical zero independent of the register clock. The RESET function occurs automatically on power-up.

Output Register Configuration

The four different register types shown in Figure 5 are described below.

D- or T-type Flip-Flops

When either a D- or T-type Flip-Flop is configured as part of the I/O structure, all eight of the product terms into the Macrocell are ORed together and fed into the register input.

JK or SR Registers

When either a JK or SR register is configured, the eight product terms are shared among two OR gates (one for the J or S input and the other for the K or R input). The allocation for these product terms for each of the register inputs is optimized by the iPLDS II development software.

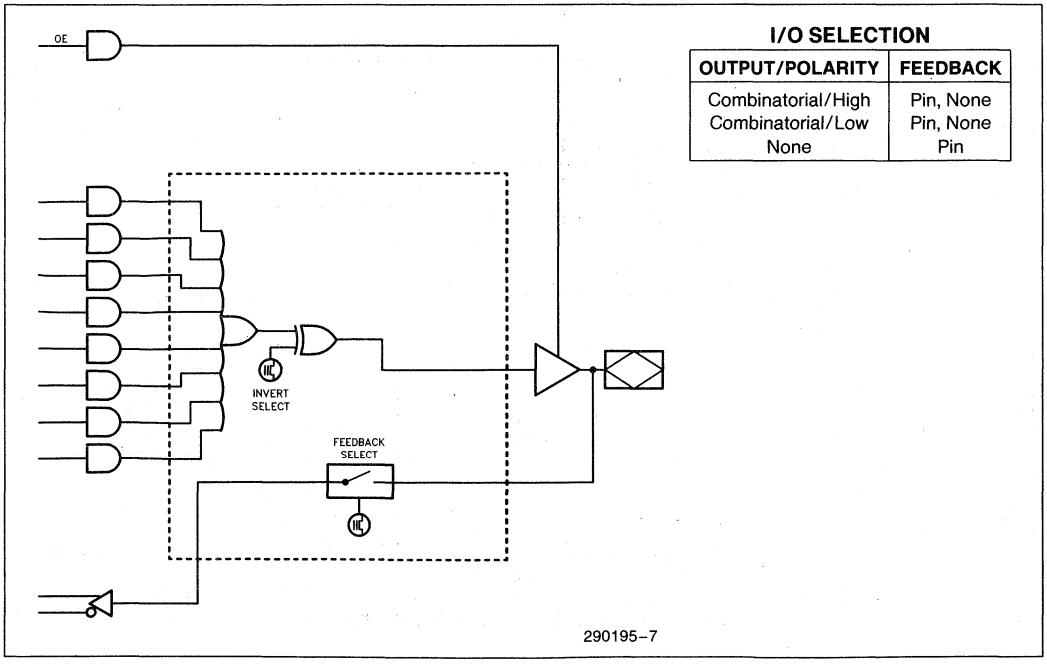
OUTPUT/FEEDBACK

The Output Select Multiplexer allows for either registered, combinatorial or no output.

The Feedback Select Multiplexer EPROM bit enables registered, I/O (using the pin for bidirectional input or just input), or no feedback to the AND array.

The Feedback Select is also important for building product terms with more than 8 products. The 8-product product term of a Macrocell can be fed back into the AND array and combined with still more signals to create a much larger product term (of more than 8-inputs). In addition, if the feedback product term is not to be output, then the iPLDS II will reserve the associated Macrocell pin and indicate it in the REPORT file. A reserved pin should be left floating (no connect) when assembled onto a circuit board.

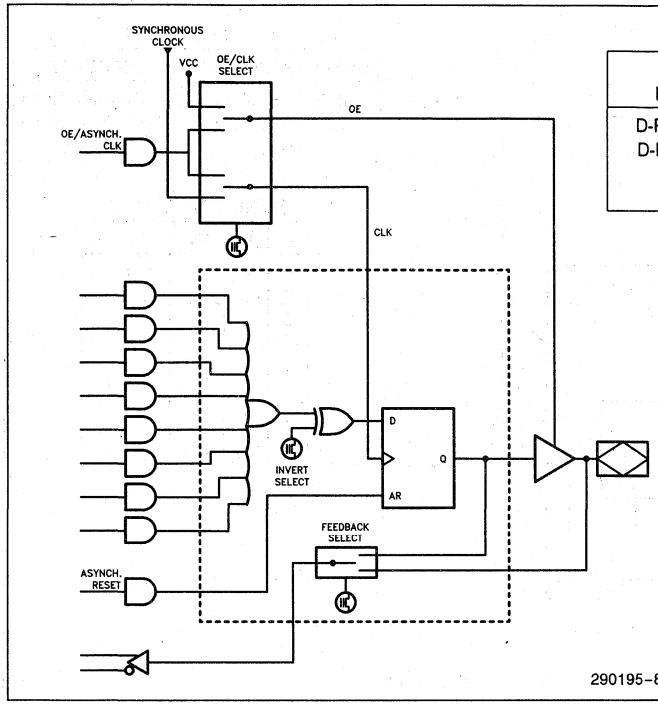
Any I/O pin may be configured as a dedicated input by selecting no output and pin feedback through the appropriate multiplexers.



I/O SELECTION

OUTPUT/POLARITY	FEEDBACK
Combinatorial/High	Pin, None
Combinatorial/Low	Pin, None
None	Pin

Figure 5a. Combinatorial I/O Configuration



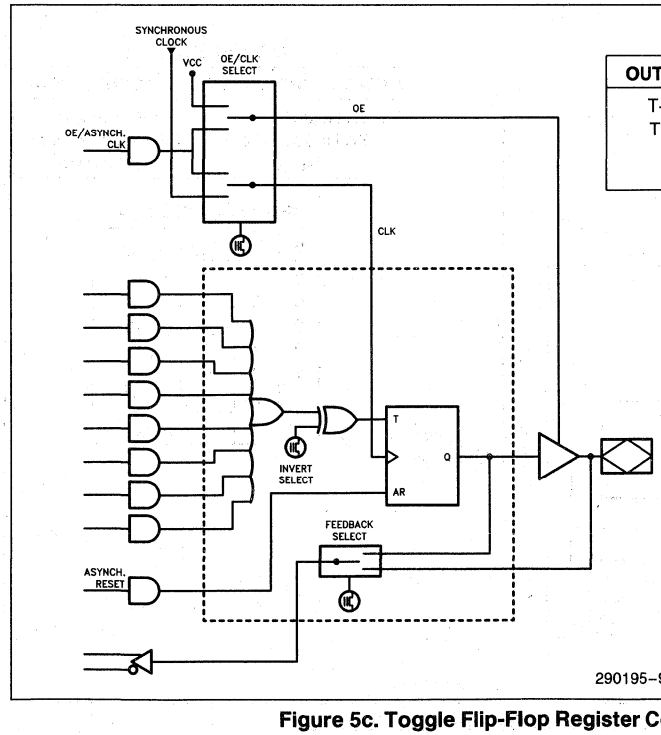
I/O SELECTION

OUTPUT/ POLARITY	FEEDBACK
D-Register/High	D-Register, Pin, None
D-Register/Low	D-Register, Pin, None
None	D-Registered
None	Pin

FUNCTION TABLE

D	Q_n	$Q_n + 1$
0	0	0
0	1	0
1	0	1
1	1	1

Figure 5b. D-Type Flip-Flop Register Configuration



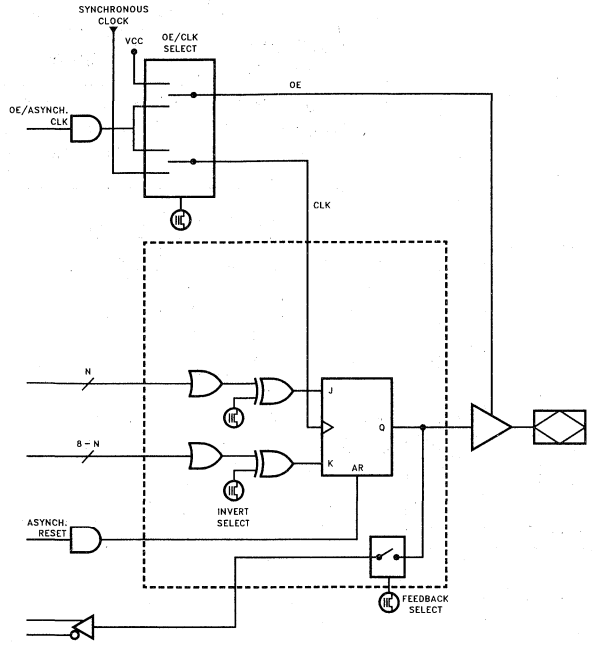
I/O SELECTION

OUTPUT/POLARITY	FEEDBACK
T-Register/High	T-Register, Pin, None
T-Register/Low	T-Register, Pin, None
None	T-Register
None	Pin

FUNCTION TABLE

T	Q_n	$Q_n + 1$
0	0	0
0	1	1
1	0	1
1	1	0

Figure 5c. Toggle Flip-Flop Register Configuration



290195–10

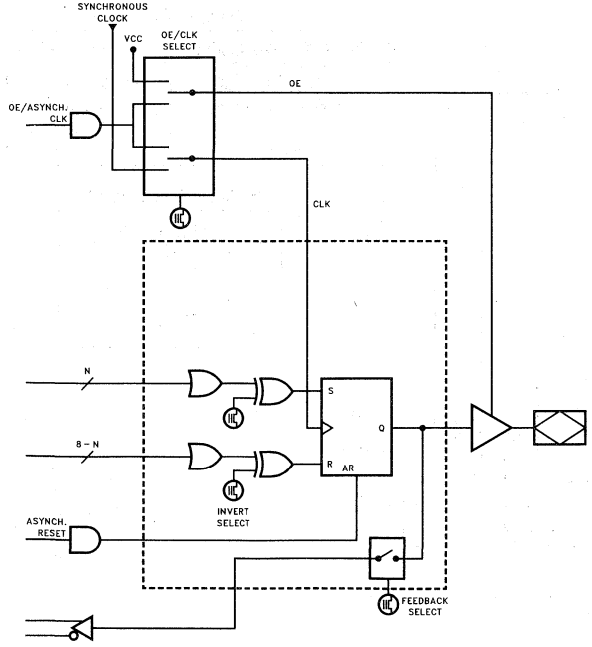
I/O SELECTION

OUTPUT/POLARITY	FEEDBACK
JK Register/High	JK Register, None
JK Register/Low	JK Register, None
None	JK Register

FUNCTION TABLE

J	K	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Figure 5d. JK Flip-Flop Register Configuration



290195–11

I/O SELECTION

OUTPUT/POLARITY	FEEDBACK
SR Register/High	SR Register, None
SR Register/Low	SR Register, None
None	SR Register

FUNCTION TABLE

S	R	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	Illegal	

Figure 5e. SR Flip-Flop Register Configuration

Erased-State Configuration

Prior to programming or after erasing, the I/O structure is configured for combinatorial active low output with input (pin) feedback.

ERASURE CHARACTERISTICS

Erasure characteristics of the device are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000Å. It should be noted that sunlight and certain types of flourescent lamps have wavelengths in the 3000–4000Å. Data shows that constant exposure to room level flourescent lighting could erase the typical device in approximately three years, while it would take approximately one week to cause erasure when exposed to direct sunlight. If the 5C090 is to be exposed to these types of lighting conditions for extended periods of time, conductive opaque labels should be placed over the device window to prevent unintentional erasure.

The recommended erasure procedure for the 5C090 is exposure to shortwave ultraviolet light with a wavelength of 2537Å. The integrated dose (i.e., UV intensity × exposure time) for erasure should be a minimum of fifteen (15) Wsec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12,000 μW/cm^2 power rating. The 5C090 should be placed within one inch of the lamp tubes during erasure. The maximum integrated dose the 5C090 can be exposed to without damage is 7258 Wsec/cm^2 (1 week at 12,000 μW/cm^2). Exposure to high intensity UV light for longer periods may cause permanent damage to the device.

PROGRAMMING CHARACTERISTICS

Initially, and after erasure, all the EPROM control bits of the 5C090 are connected (in the "1" state). Each of the connected control bits are selectively disconnected by programming the EPROM cells into their "0" state. Programming voltage and waveform specifications are available by request from Intel to support programming of the 5C090.

int$_e$ligent Programming™ Algorithm

The 5C090 supports the int$_e$ligent Programming Algorithm which rapidly programs Intel ELPDs using an efficient and reliable method. The int$_e$ligent Programming Algorithm is particularly suited to the production programming environment. This method ensures reliability as the incremental program margin of each bit is continually monitored to determine when the bit has been successfully programmed.

FUNCTIONAL TESTING

Since the logical operation of the 5C090 is controlled by EPROM elements, the device is completely testable. Each programmable EPROM bit controlling the internal logic is tested using application-independent test program patterns. After testing, the devices are erased before shipment to customers. No post-programming tests of the EPROM array are required.

The testability and reliability of EPROM-based programmable logic devices is an important feature over similar devices based on fuse technology. Fuse-based programmable logic devices require a user to perform post-programming tests to insure proper programming. These tests must be done at the device level because of the cummulative error effect. For example, a board containing ten devices each possessing a 2% device fallout translates into an 18% fallout at the board level (it should be noted that programming fallout of fuse-based programmable logic devices is typically 2% or higher).

DESIGN RECOMMENDATIONS

For proper operation, it is recommended that all input and output pins be constrained to the voltage range GND < (V_{IN} or V_{OUT}) < V_{CC}. Unused inputs and I/Os should be tied to V_{CC} or GND to minimize device power consumption. Reserved pins (as indicated in the logic compiler REPORT file) should be left floating (no connect) so that the pin can attain the appropriate logic level. A power supply decoupling capacitor of at least 0.2 μF must be connected directly between V_{CC} and GND pins of the device.

As with all CMOS devices, ESD handling procedures should be used with the 5C090 to prevent damage to the device during programming, assembly and test.

DESIGN SECURITY

A single EPROM bit provides a programmable design security feature that controls the access to the data programmed into the device. If this bit is set, a proprietary design within the device cannot be copied. This EPROM security bit enables a higher de-

gree of design security than fused-based devices since programmed data within EPROM cells is invisible even to microscopic evaluation. The EPROM security bit, along with all the other EPROM control bits, will be reset by erasing the device.

AUTOMATIC STAND-BY MODE

The 5C090 contains a programmable bit, the Turbo Bit, that optimizes operation for speed or for power savings. When the Turbo Bit is programmed (TURBO = ON), the device is optimized for maximum speed. When the Turbo Bit is not programmed (TURBO = OFF), the device is optimized for power savings by entering standby mode during periods of inactivity.

Figure 6 shows the device entering standby mode approximately 100 ns after the last input transition. When the next input transition is detected, the device returns to active mode. Wakeup time adds an additional 25 ns to the propagation delay through the device as measured from the first input. No delay will occur if an output is dependent on more than one input and the last of the inputs changes after the device has returned to active mode.

After erasure, the Turbo Bit is unprogrammed (OFF); automatic standby mode is enabled. When the Turbo Bit is programmed (ON), the device never enters standby mode.

LATCH-UP IMMUNITY

All of the input, I/O, and clock pins of the 5C090 have been designed to resist latch-up which is inherent in inferior CMOS structures. The 5C090 is designed with Intel's proprietary CHMOS II-E EPROM process. Thus, each of the pins will not experience latch-up with currents up to ±100 mA and voltages ranging from −1V to (V_{CC} + 1V). Furthermore, the programming pin is designed to resist latch-up to the 13.5V maximum device limit.

INTEL PROGRAMMABLE LOGIC DEVELOPMENT SYSTEM II (iPLDS II)

iPLDS II provides all the tools needed to design with Intel EPLDs or compatible devices. In addition to providing development assistance, iPLDS II insulates the user from having to know all the intricate details of EPLD architecture (the machine will optimize a design to benefit from architectural features). It contains comprehensive third generation software that supports four different design entry methods, minimizes logic, does automatic pin assignments and produces the best design fit for the selected EPLD. It is user friendly with guided menus, on-line Help messages and soft key inputs.

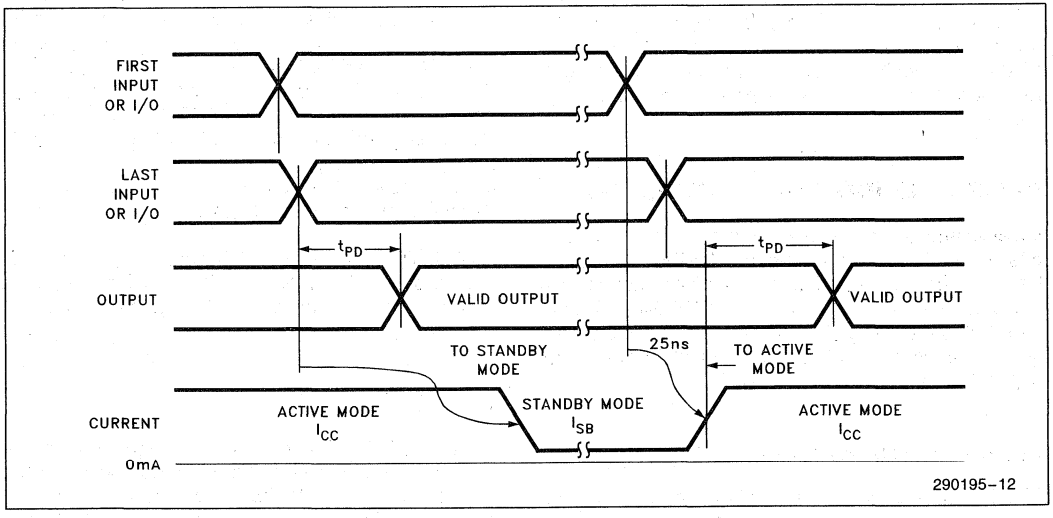


Figure 6. 5C090 Standby and Active Mode Transitions

In addition, the iPLDS II contains programmer hardware in the form of an iUP-PC Universal Programmer Personal Computer to enable the user to program EPLDs, read and verify programmed devices and also to graphically edit programming files. The software generates industry standard JEDEC object code output files which can be downloaded to other programmers as well.

iPLS II software interfaces to several schematic capture packages to enable designs to be entered in schematic form. IPLDview-286/IPLDdraw allows the designer to use familiar TTL symbols or EPLD design primitive symbols. User-defined symbols are also supported. IPLDdraw also provides a path to A.C. Timing simulation of EPLD designs.

SCHEMA III-PLD allows the designer to use TTL symbols, EPLD custom macros, or EPLD design primitive symbols. It also supports user-defined symbols.

Other design formats include Boolean equation entry (supported directly by iPLS II) and state machine entry (supported by iSTATE).

Detailed information on the Intel Programmable Logic Development System II is contained in a separate Intel data sheet. (Order Number: 280168). Refer to the tools section of the *Programmable Logic* handbook for complete information on development tools.

The 5C090 is also supported by third-party logic compilers such as ABEL*, CUPL*, PLDesigner, Log/IC, etc. Programming support is provided by third-party programmer companies such as Data I/O, Logical Devices, STAG, etc. Please refer to the "Third-Party Support" lists in the *Programmable Logic* handbook for complete information and vendor contacts.

ADF PRIMITIVES SUPPORTED

The following ADF primitives are supported by this device:

INP	JOJF
CONF	JONF
COIF	SONF
RONF	SOSF
RORF	TOIF
ROIF	TONF
NORF	TOTF
NOJF	CLKB
NOSF	
NOTF	

ORDERING INFORMATION

t_{PD} (ns)	t_{CO} (ns)	f_{MAX} (MHz)	Order Code	Package	Operating Range
50	23	26.3	D5C090-50	CERDIP	Commercial
			P5C090-50	PDIP	
			N5C090-50	PLCC	
60	25	21.7	D5C090-60	CERDIP	Commercial
			P5C090-60	PDIP	
			N5C090-60	PLCC	

*ABEL is a trademark of Data I/O, Corporation. CUPL is a trademark of Logical Devices, Inc. PLDesigner is a trademark of MINC, Inc.

ABSOLUTE MAXIMUM RATINGS*

Symbol	Parameter	Min	Max	Units
V_{CC}	Supply Voltage(1)	−2.0	7.0	V
V_{PP}	Programming Supply Voltage(1)	−2.0	13.5	V
V_I	DC Input Voltage(1)(2)	−0.5	V_{CC}+0.5	V
t_{stg}	Storage Temperature	−65	+150	°C
t_{amb}	Ambient Temperature(3)	−10	+85	°C

NOTES:
1. Voltages with respect to ground.
2. Minimum DC input is −0.5V. During transitions, the inputs may undershoot to −2.0V or overshoot to +7.0V for periods less than 20 ns under no load conditions.
3. Under bias. Extended temperature versions are also available.

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Min	Max	Unit
V_{CC}	Supply Voltage	4.75	5.25	V
V_{IN}	Input Voltage	0	V_{CC}	V
V_O	Output Voltage	0	V_{CC}	V
T_A	Operating Temperature	0	+70	°C
t_R	Input Rise Time		500	ns
t_F	Input Fall Time		500	ns

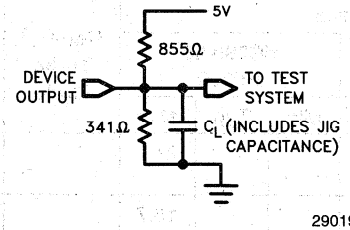
NOTE:
4. t_R, t_F for CLK is 250 ns max.

D.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5.0V ±5%

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
V_{IH}(5)	HIGH Level Input Voltage			2.0		V_{CC} + 0.3	V
V_{IL}(5)	LOW Level Input Voltage			−0.3		0.8	V
V_{OH}(6)	HIGH Level Output Voltage	I_O = −4.0 mA DC, V_{CC} = Min.		2.4			V
V_{OL}	LOW Level Output Voltage	I_O = 4.0 mA DC, V_{CC} = Min.				0.45	V
I_I	Input Leakage Current	V_{CC} = Max., GND < V_{IN} < V_{CC}				±10.0	µA
I_{OZ}	Output Leakage Current	V_{CC} = Max., GND < V_{OUT} < V_{CC}				±10.0	µA
I_{SC}(7)	Output Short Circuit Current	V_{CC} = Max., V_{OUT} = 0.5V			20	30	mA
I_{SB}(8)	Standby Current (Standby)	V_{CC} = Max., V_{IN} = V_{CC} or GND			50	100	µA
I_{CC}	Power Supply Current (Active) (Turbo Bit Off) Device Prog. as Two 12-Bit Ctrs. (See I_{CC} vs. Freq. Graph)	V_{CC} = Max., V_{IN} = V_{CC} or GND	No Load, Input Freq. = 1 MHz		15	25	mA

NOTES:
5. Absolute values with respect to device GND; all over and undershoots due to system or tester noise are included.
6. I_O at CMOS levels (3.84V) = −2 mA.
7. Not more than 1 output should be tested at a time. Duration of that test must not exceed 1 second.
8. With Turbo Bit Off, device automatically enters standby mode approximately 100 ns after last input transition.

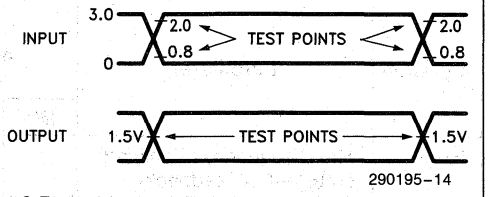
A.C. TESTING LOAD CIRCUIT



290195-13

C_L = 50 pF

A.C. TESTING INPUT, OUTPUT WAVEFORM



290195-14

A.C. Testing: Inputs are Driven at 3.0V for a Logic "1" and 0V for a Logic "0". Timing Measurements are made at 2.0V for a Logic "1" and 0.8V for a Logic "0" on inputs. Outputs are measured at a 1.5V point. Device input rise and fall times < 6 ns.

CAPACITANCE

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
C_{IN}	Input Capacitance	V_{IN} = 0V, f = 1.0 MHz			20	pF
C_{OUT}	Output Capacitance	V_{OUT} = 0V, f = 1.0 MHz			20	pF
C_{CLK}	Clock Pin Capacitance	V_{IN} = 0V, f = 1.0 MHz			20	pF
C_{VPP}	V_{PP} Pin	CLK2 on 5C090, f = 1.0 MHz			80	pF

A.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = 5V ±5%, Turbo Bit On[9]

Symbol	From	To	Device						Non-[11] Turbo Mode	Unit
			5C090-50 EP900-2			5C090-60 EP900				
			Min	Typ	Max	Min	Typ	Max		
t_{PD1}	Input	Comb. Output			45			55	+25	ns
t_{PD2}	I/O	Comb. Output			50			60	+25	ns
t_{PZX}[10]	I or I/O	Output Enable			50			60	+25	ns
t_{PXZ}[10]	I or I/O	Output Disable			50			60	+25	ns
t_{CLR}	Asynch. Reset	Q Reset			50			60	+25	ns

NOTES:
9. Typical Values are at T_A = 25°C, V_{CC} = 5V, Active Mode.
10. t_{PZX} and t_{PXZ} are measured at ±0.5V from steady state voltage as driven by spec. output load. t_{PXZ} is measured with C_L = 5 pF.
11. If device is operated with Turbo Bit Off (Non-Turbo Mode) and the device has been inactive for approx. 100 ns, increase time by amount shown.

SYNCHRONOUS CLOCK MODE A.C. CHARACTERISTIC

T_A = 0°C to 70°C, V_{CC} = 5.0V ±5%, Turbo Bit On[9]

Symbol	Parameter	Device						Non-[11] Turbo Mode	Unit
		5C090-50 EP900-2			5C090-60 EP900				
		Min	Typ	Max	Min	Typ	Max		
f_{MAX}	Max. Frequency (Pipelined) ($1/t_{SU}$—No Feedback)			26.3			21.7		MHz
f_{CNT}	Max. Count Frequency ($1/t_{CNT}$—With Feedback)			20			16.7		MHz
t_{SU1}	Input Setup Time to CLK	36			43			+25	ns
t_{SU2}	I/O Setup Time to CLK	38			46			+25	ns
t_H	I or I/O Hold after CLK High	0			0				ns
t_{CO}	CLK High to Output Valid			23			25		ns
t_{CNT}	Register Output Feedback to Register Input—Internal Path	50			60			+25	ns
t_{CH}	CLK High Time	17.5			23				ns
t_{CL}	CLK Low Time	17.5			23				ns

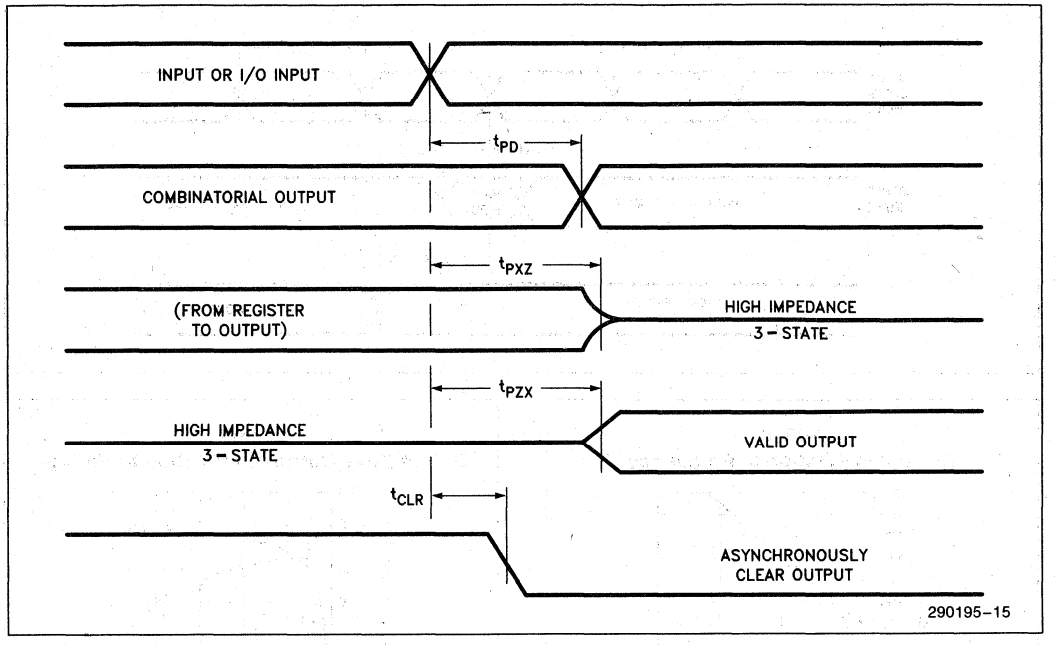
ASYNCHRONOUS CLOCK MODE A.C. CHARACTERISTICS

T_A = 0°C to 70°C, V_{CC} = 5.0V ±5%, Turbo Bit On[8]

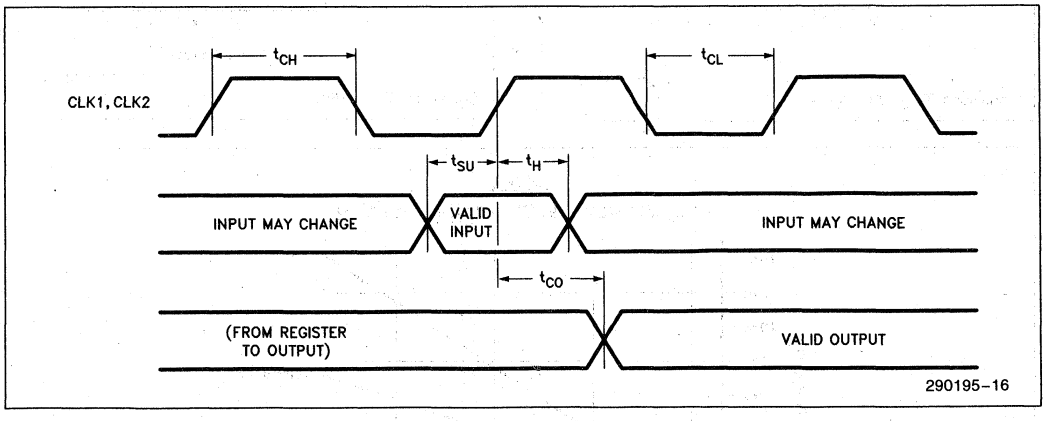
Symbol	Parameter	Device						Non-[11] Turbo Mode	Unit
		5C090-50 EP900-2			5C090-60 EP900				
		Min	Typ	Max	Min	Typ	Max		
f_{ACNT}	Max. Count Frequency ($1/t_{ACNT}$—With Feedback)			20			16.7		MHz
t_{ASU1}	Input Setup Time to Asynch. Clock	10			10			+25	ns
t_{ASU2}	I/O Setup Time to Asynch. Clock	13			15			+25	ns
t_{AH}	Input or I/O Hold After Asynch. Clock	15			15				ns
t_{ACO}	Asynch. CLK to Output Valid			48			59	+25	ns
t_{ACNT}	Register Output Feedback to Register Input—Internal Path	50			60			+25	ns
t_{ACH}	Asynch. CLK High Time	17.5			23			+25	ns
t_{ACL}	Asynch. CLK Low Time	17.5			23			+25	ns

SWITCHING WAVEFORMS

COMBINATORIAL MODE

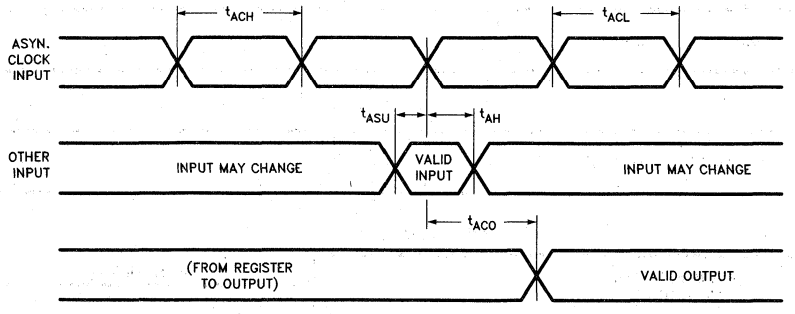


SYNCHRONOUS CLOCK MODE



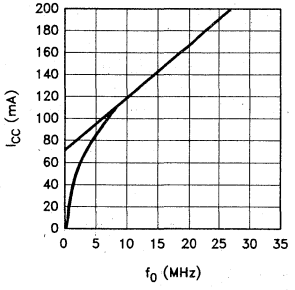
SWITCHING WAVEFORMS (Continued)

ASYNCHRONOUS CLOCK MODE



290195-17

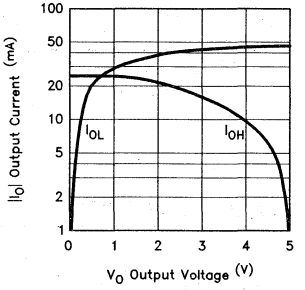
5C090
Current in Relation to Frequency



290195-18

Conditions: T_A = 0°C, V_{CC} = 5.25V

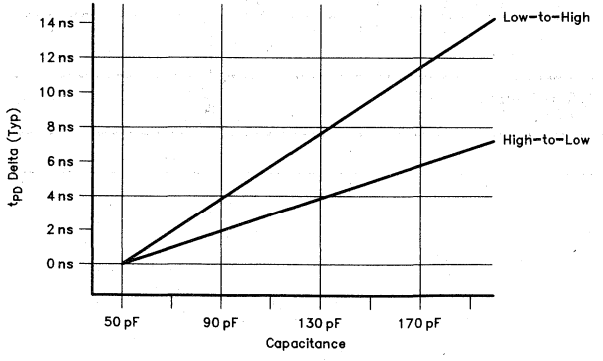
5C090
Output Drive Current in Relation to Voltage



290195-19

Conditions: T_A = 25°C

5C090
t_{PD} Derating vs. Capacative Loading



290195-20

CONDITIONS:
T_A = 25°C, V_{CC} = 5.0V

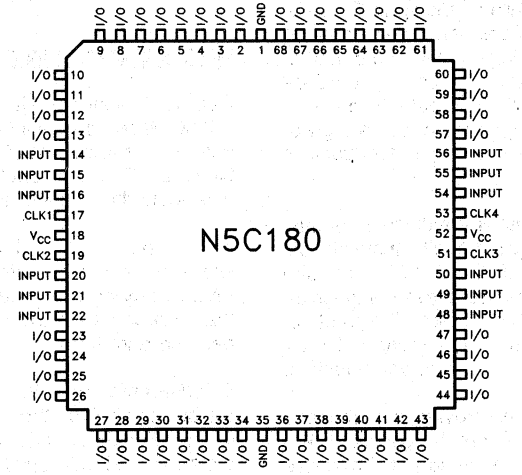
5C180
48-MACROCELL CHMOS EPLD

- **High-Performance LSI Semicustom Logic Alternative for TTL and 74HC SSI and MSI Logic**
- **48 Macrocells with Programmable I/O Architecture; up to 64 Inputs (16 Dedicated, 48 I/O) or 48 Outputs**
- **High Speed t_{PD} (max) 70 ns, 20.8 MHz Pipelined, 16.1 MHz w/Feedback**
- **Dual Feedback Signals Allowing I/O Pins to Be Used for Buried Logic and Dedicated Input**
- **Programmable Clock System with Four Synchronous Clocks as well as Asynchronous Clocking Option on All Registers**
- **Programmable Registers. Can Be Configured as D, T, SR or JK Types with Individual Reset Controls**
- **Low Power; 100 μW Typical Standby Dissipation**
- **Programmable "Security Bit" Allows Total Protection of Proprietary Designs**
- **100% Generically Tested Logic Array**
- **68-Pin J-Lead Chip Carrier**
  (See packaging spec., Order #231369)
- **100% Compatible with EP1800**

2

The Intel 5C180 EPLD (Erasable Programmable Logic Device) is a CHMOS, 48-macrocell, general-purpose PLD. This user-customizable Logic Device is available in a 68-pin PLCC package and has the benefits of low power and increased flexibility.

The 5C180 EPLD uses CHMOS EPROM (floating gate) cells as logic control elements instead of fuses. Use of Intel's advanced CHMOS II-E EPROM process technology enables greater logic densities to be achieved with superior speed and power performance. The EPROM technology also enables these devices to be 100% factory tested by the programming and the erasure of all the EPROM logic control elements in the device.



290111-1

Figure 1. Pin Configurations

September 1990
Order Number: 290111-007

The architecture of the 5C180 is based on the "Sum of Products" PLA (Programmable Logic Array) structure with a programmable AND array feeding into a fixed OR array. The 48 macrocells of the 5C180 can be partitioned into 4 identical quandrants each containing 12 macrocells. This device makes use of a segmented PLA structure with local and global bus structures to provide for increased performance and greater device utilization. The 5C180 has unique architectural features that allow programming of all 48 registers to D, T, SR or JK configurations without sacrificing product terms. These registers can be either clocked asynchronously or in banks with four synchronous clocks. In addition, the 16 global macrocells have two independent feedback paths to the array that allow for buried logic implementation together with use of the I/O pin for input functions.

ARCHITECTURE DESCRIPTION

Externally, the 5C180 provides 12 dedicated data inputs, 4 synchronous clock inputs, and 48 I/O pins which may be individually programmed for input, output, or bi-directional operation.

The Block Diagram is shown in Figure 2 with pin numbers for the PLCC package. The internal architecture is organized in familiar sum-of-products (AND-OR) structure. The 5C180 houses a total of 480 product terms distributed among 48 Macrocells. The basic Macrocell structure is shown in Figure 3. Input and feedback signals are selectively connected to product terms via EPROM cells. The output of the AND array feeds a fixed OR gate to produce sum-of-products logic. The final output may be combinatorial or registered, programmed active high or low. Combinatorial, registered, or pin feedback is also user-defined.

The 5C180 is partitioned into 4 identical quadrants. Each quadrant contains 12 Macrocells. Input signals to the Macrocells come from the 5C180 Local and Global bus structures. These two buses comprise an 88-input AND array for each quadrant. The output of each Macrocell feeds an I/O Architecture Control Block which contains output and feedback selection.

Four dedicated clock inputs provide synchronous clock signals to the 5C180 internal registers. There is one synchronous clock per quadrant. Therefore each clock signal controls a bank of 12 registers. CLK1 may be connected to registers in Macrocells 1–12, CLK2 with Macrocells 13–24, CLK3 with Macrocells 25–36, and CLK4 with Macrocells 37–48. With synchronous clocks, the flip-flops are positive edge triggered. Both true and complement signals for each dedicated clock input may also be used within the AND array. All 48 internal registers may be individually programmed for synchronous or asynchronous clocking. Asynchronous clocking is possible via a Macrocell product term. Clock inputs not used for synchronous clock signals may be used as global bus inputs.

Invert Select EPROM Bit

The Invert Select EPROM bit is used to invert the product term input into the register. This applies to all inputs including double inputs on JK and SR registers. The invert option allows the highest possible logic utilization by use of deMorgan logic inversion.

At each intersecting point in the logic array there exists an EPROM-type programmable connection. Initially, all connections are complete. This means that both the true and complement of all inputs are connected to each product term. Connections are opened during the programming process. Therefore any product term can be connected to the true or complement of any input. When both the true and complement connections of any input are left intact, a logical false results on the output of the AND gate. If both the true and complement connections of any input are programmed open, then a logical "don't care" results for that input. If all inputs for a product term are programmed open, then a logical true results on the output of the AND gate.

BUS STRUCTURE

Input and feedback signals are connected to each 5C180 Macrocell via a Local and Global Bus. Figure 4 shows the Macrocell-Bus interface for Quadrant D. The Global Bus contains 64 input signals while the Local Bus has 24.

Within the 5C180 Macrocell, the product-terms share the entire bus structure. Therefore, a logical AND of any of the variables (or their complements) that is present on the buses may be produced by each product term.

All quadrants share the same Global Bus. Inputs to the bus come from the true and complement signals of the 12 dedicated data inputs, 4 clock inputs, and the 16 Global Macrocell pin feedback signals.

Each quadrant has its own Local Bus. Inputs to this bus come from the 12 quadrant Macrocells. For the eight Local Macrocells, the signals can be either from the Macrocell internal logic or from the pin. For the four Global Macrocells, the signals come from the Macrocell internal logic only.

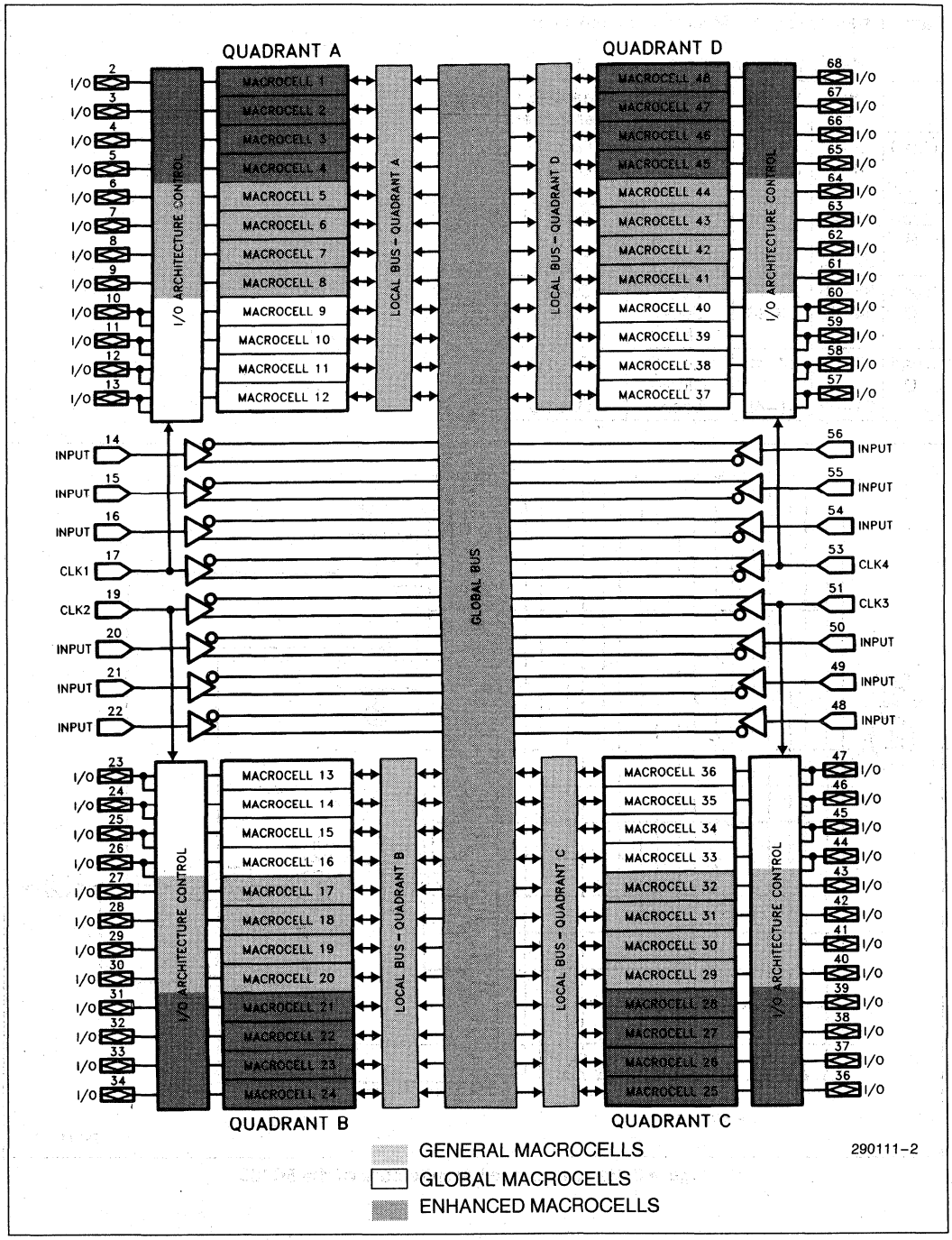


Figure 2. 5C180 Block Diagram—PLCC Package

Table 1 summarizes the Macrocell interconnect.

Table 1. Macrocell Interconnect

	Pin #	Macro-cell #	Feedback Structure	Feedback Interconnect
Quad A	2–9	1–8	Local	Quad A
	10–13	9–12	Local	Quad A
			Global	All
Quad B	23–26	13–16	Local	Quad B
			Global	All
	27–34	17–24	Local	Quad B
Quad C	36–43	25–32	Local	Quad C
	44–47	33-36	Local	Quad C
			Global	All
Quad D	57–60	37–40	Local	Quad D
			Global	All
	61–68	41–48	Local	Quad D

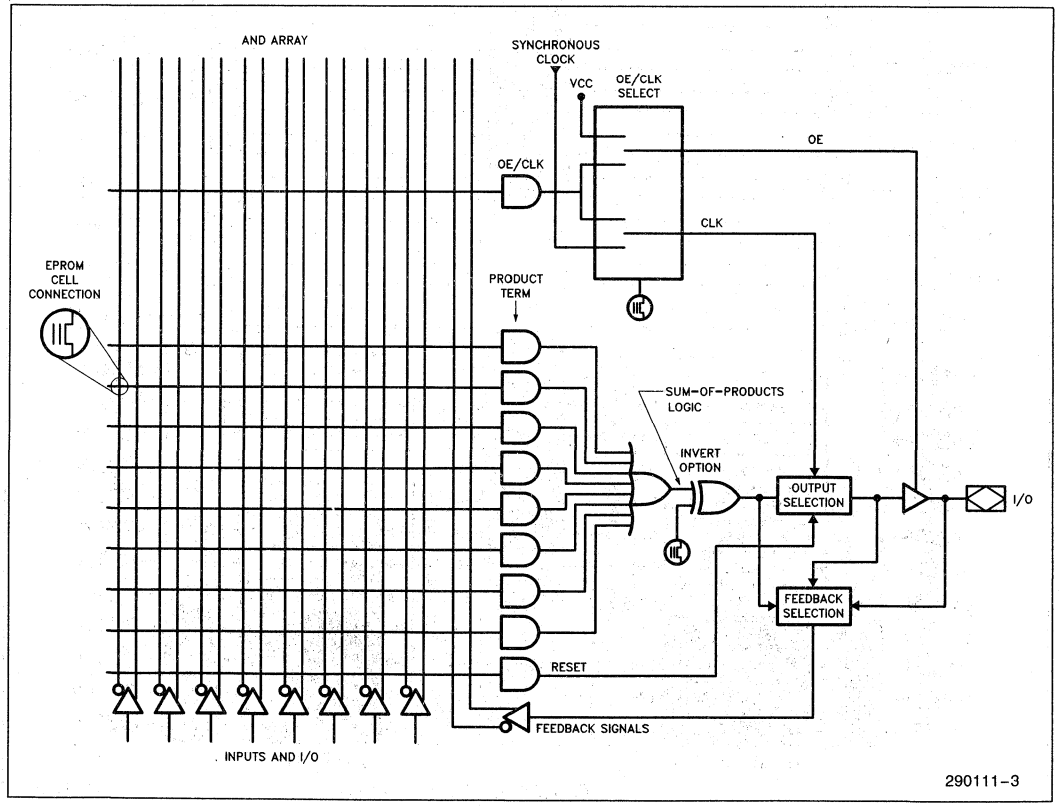


Figure 3. Basic Macrocell Architecture of the 5C180

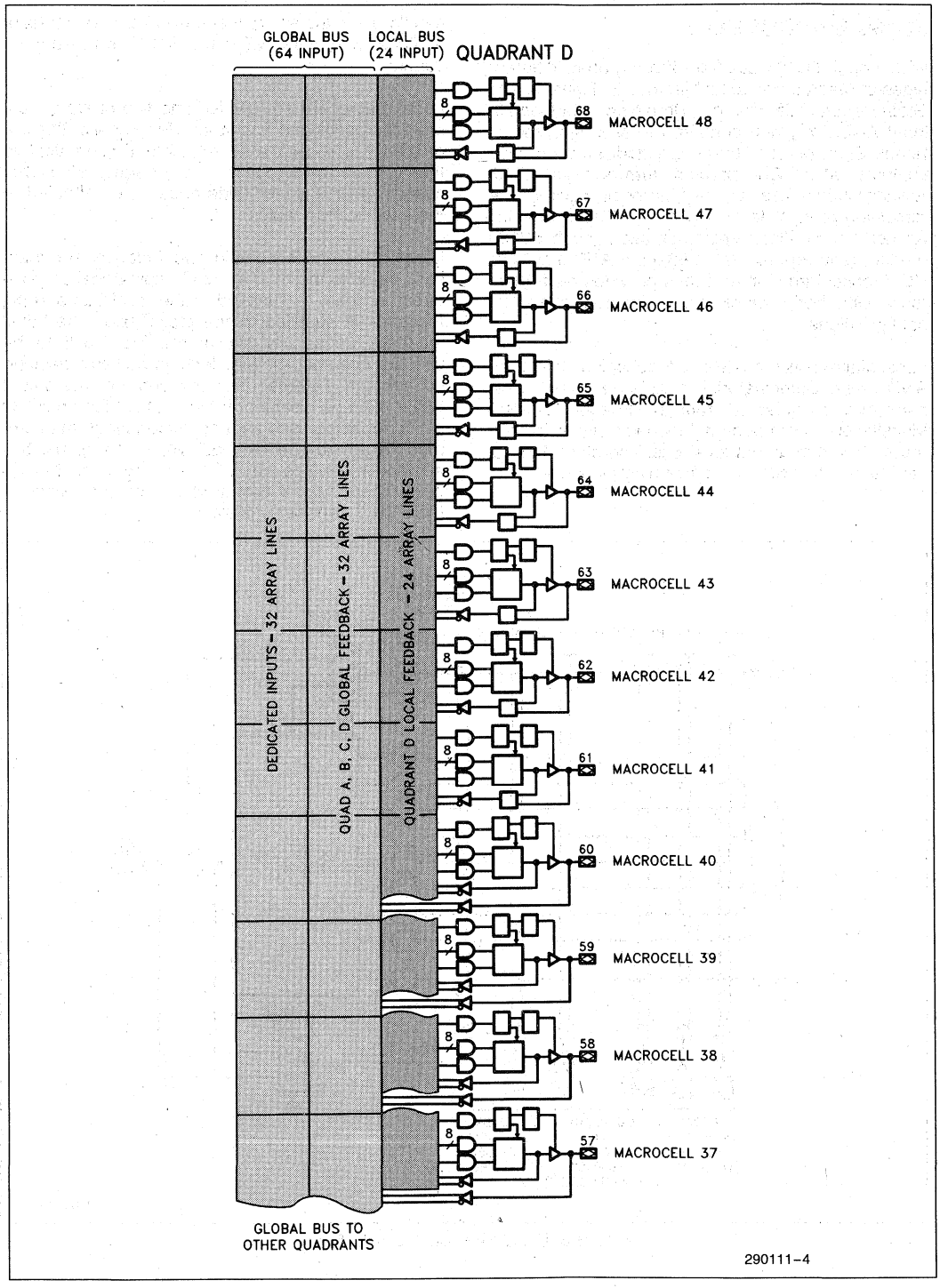


Figure 4. Quadrant "D" Bus Interface

5C180 MACROCELLS

Within each 5C180 quadrant there are two different types of Macrocells; Local Macrocells, Figure 5, and Global Macrocells, Figure 6. Both types share an 88-input AND array and contain a total of ten product terms. Eight product terms are dedicated for logic implementation. One product term is reserved for Asynchronous Clear to the Macrocell register. The remaining product term is used for Output Enable/ Asynchronous Clock implementation. Each 5C180 product term represents an 88-input AND gate. The I/O Architecture Control Block provides each Macrocell with both combinatorial and registered I/O configurations.

Local Macrocells provide one feedback path into the AND array. Combinatorial, registered or pin feedback may be selected from the Feedback Select Multiplexer. The selected feedback signal is then routed to the quadrant local bus. Therefore, the Local Macrocell feedback communicates only to Macrocells within the same quadrant. There are a total of 32 Local Macrocells within the 5C180, with eight per quadrant.

Local macrocells are divided into two groups: General Macrocells and Enchanced Macrocells. The Enhanced Macrocells are architecturally identical to the General Macrocells but operate at higher speeds. These speed differences are reflected in the specification tables.

Global Macrocells contain two independent feedback paths to the AND array. Combinatorial or registered feedback is supplied to the local bus and pin feedback is supplied to the global bus. The "dual feedback" capability allows the Macrocell to be used for internal logic functions as well as a dedicated input pin. To obtain this configuration, the output buffer must be disabled. If the Global Macrocell I/O pin is not being used as a dedicated input, the Macrocell logic may be fed back along the global bus allowing routing to any of the 5C180's 48 Macrocells. There are 16 Global Macrocells contained in the 5C180, four per quadrant.

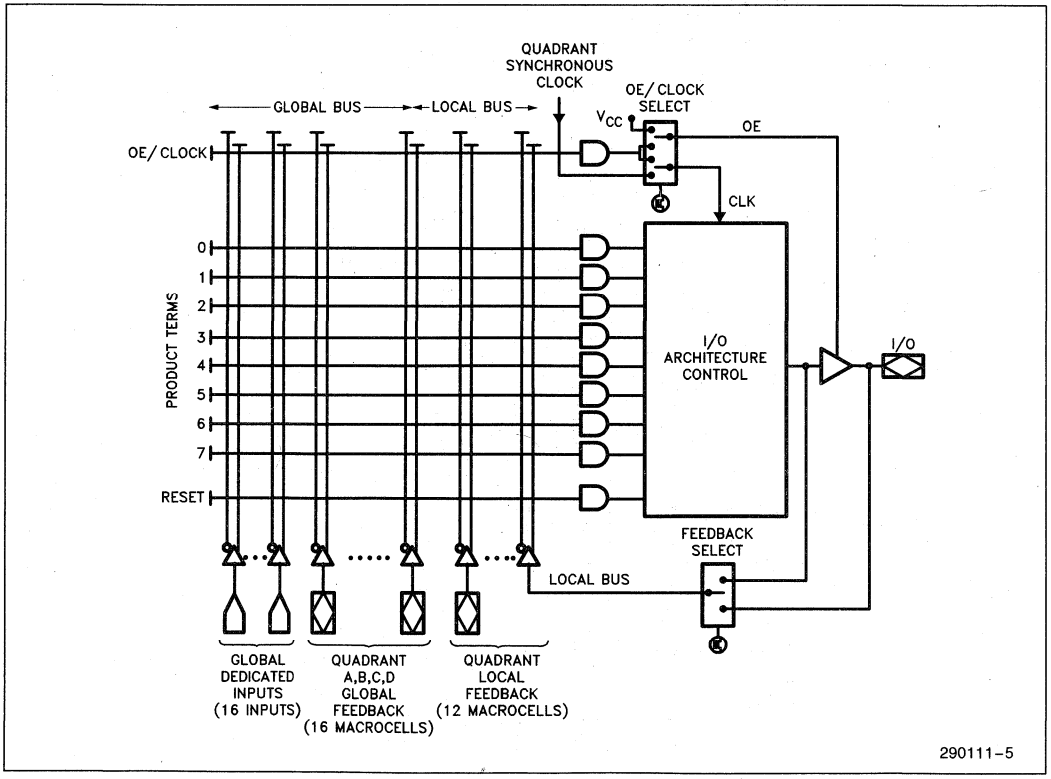


Figure 5. Local Macrocell Logic Array

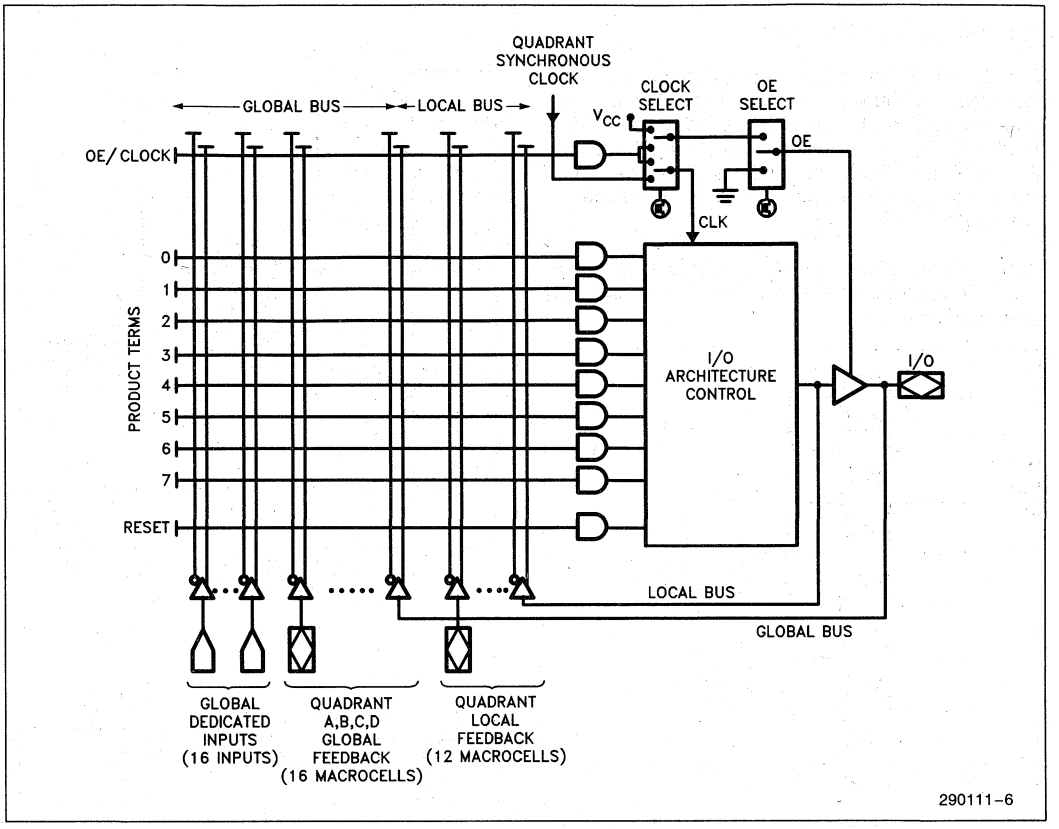


Figure 6. Global Macrocell Logic Array

MACROCELL LOGIC CONFIGURATIONS

Combinatorial Selection

In the Combinatorial configuration, eight product terms are ORed together to generate the output signal. The Invert Select EPROM bit controls output polarity and the Output Enable buffer is product-term controlled. The Feedback Select allows the user to choose combinatorial, I/O (pin) or no feedback to the respective local and global buses.

REGISTER SELECTION

The advanced I/O architecture of the 5C180 allows four different register types along with combinatorial output as illustrated in Figures 8a–8e. The register types include a T, D, JK, or SR Flip-Flop and each Macrocell I/O structure may be independently configured. In addition, all registers have an individual asynchronous RESET control from a dedicated product term derived in the AND array. When this dedicated product term is a logical one, the Macrocell register is immediately cleared to a logical zero independent of the register clock. The RESET function occurs automatically on power-up.

The four different register types shown in Figures 7b–7e are described below:

D- or T-type Flip-Flops

When either a D- or T-type Flip-Flop is configured as part of the I/O structure, all eight of the product terms into the Macrocell are ORed together and fed into the register input.

JK or SR Registers

When either a JK or SR register is configured, the eight product terms are shared among two OR gates (one for the J or S input and the other for the K or R input). The allocation for these product terms for each of the register inputs is optimized by the iPLDS II development software.

Buried Logic Selection

For Global Macrocells, if no output is selected, the logic may be "buried" and the I/O pin can be used as an additional dedicated input. The use of "dual feedback" is accomplished by tri-stating the Output Enable Buffer. Thus, up to 16 additional dedicated inputs may be added without sacrificing the Macrocell internal logic.

In the erased state, the I/O architecture is configured for combinatorial active low output with I/O (pin) feedback.

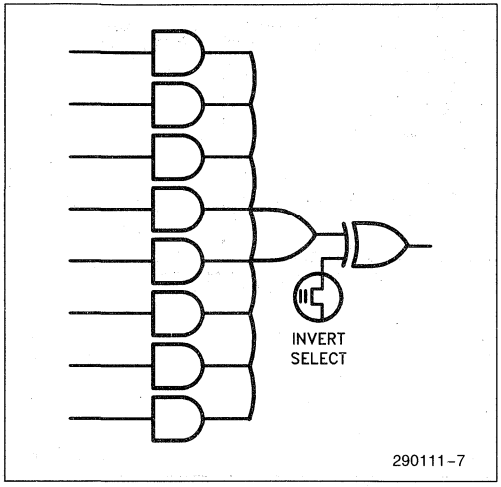


Figure 7a. Combinatorial I/O Configuration

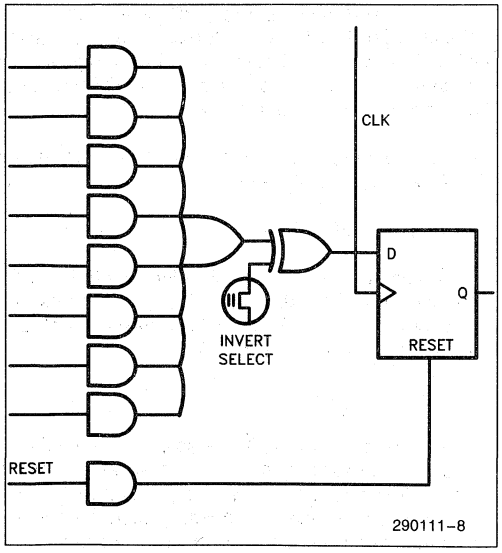


Figure 7b. D-Type Flip-Flop Register Configuration

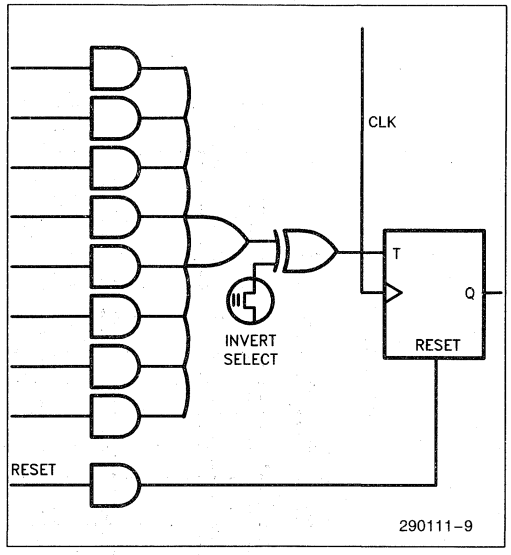


Figure 7c. Toggle Flip-Flop Register Configuration

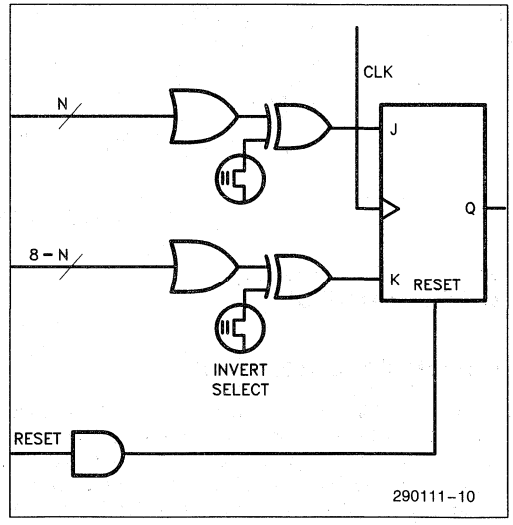


Figure 7d. JK Flip-Flop Register Configuration

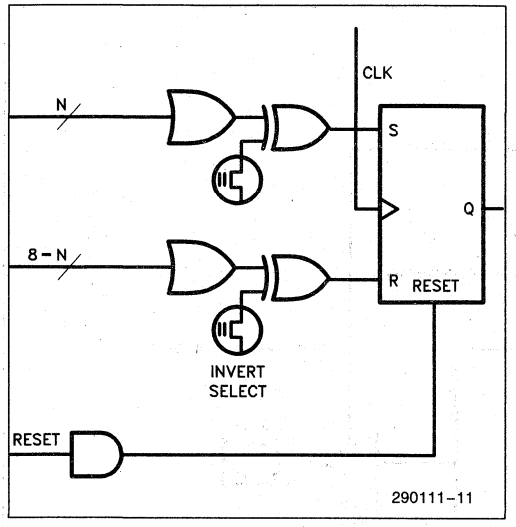


Figure 7e. SR Flip-Flop Register Configuration

MACROCELL OE/CLK SELECT

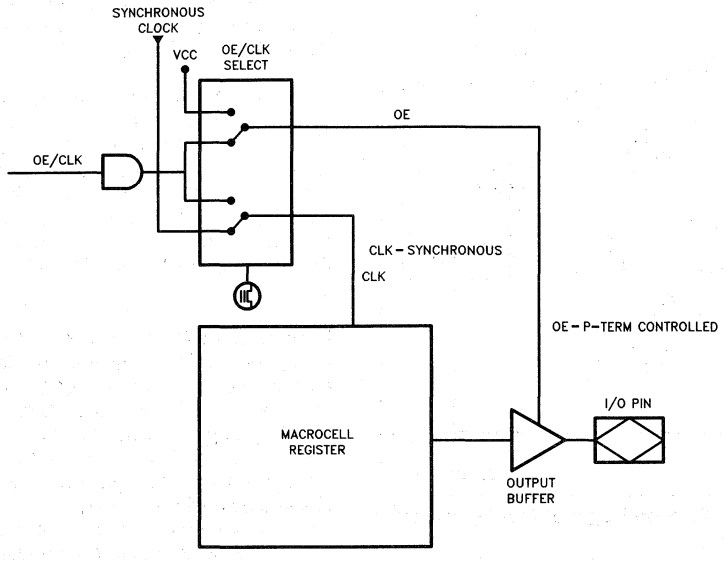
Each 5C180 register may be clocked synchronously or asynchronously. Figure 8a and 8b shows the modes of operation provided by the OE/CLK Select Multiplexers for both Local and Global Macrocells.

The operation of each multiplexer is controlled by EPROM bits and may be individually configured for each 5C180 Macrocell.

In Mode 0, the three-state output buffer is controlled by a single product term. If the output of the AND gate is a logical true then the output buffer is enabled. If a logical false resides on the output of the AND gate then the output buffer is seen as high impedance. In this mode the Macrocell flip-flop may be clocked by its quadrant synchronous clock input. In the erased state, the 5C180 is configured as Mode 0.

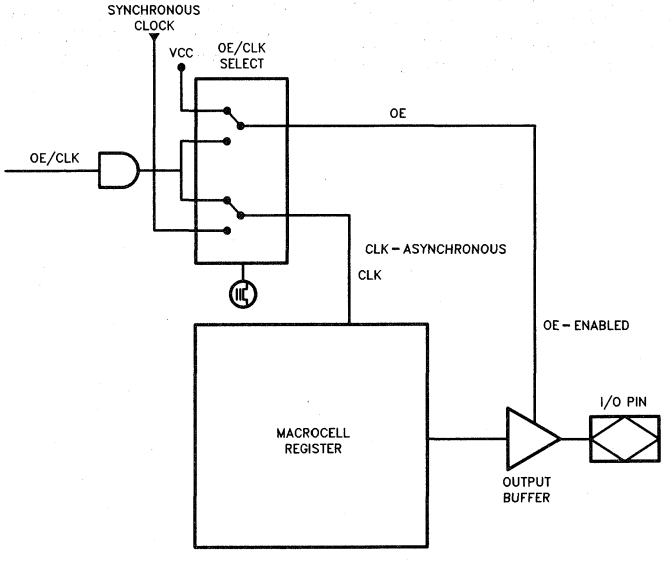
In Mode 1, the Output Buffer is always enabled. The Macrocell flip-flop now may be triggered from an asynchronous clock signal generated by the Macrocell product term. This mode allows individual clocking of flip-flops from any available signal in the quadrant AND array. Because both true and complement signals reside in the AND array, the flip-flops may be clocked by positive- or negative-going signals at any input pin. With the clock now controlled by a product term, gate clock structures are also possible.

In Modes 2 and 3, the Output Buffer is always disabled. The Macrocell flip-flop may still be triggered from clock signals generated from the Macrocell product term or asynchronous clocks. This mode is only possible for Global Macrocells.



290111–12

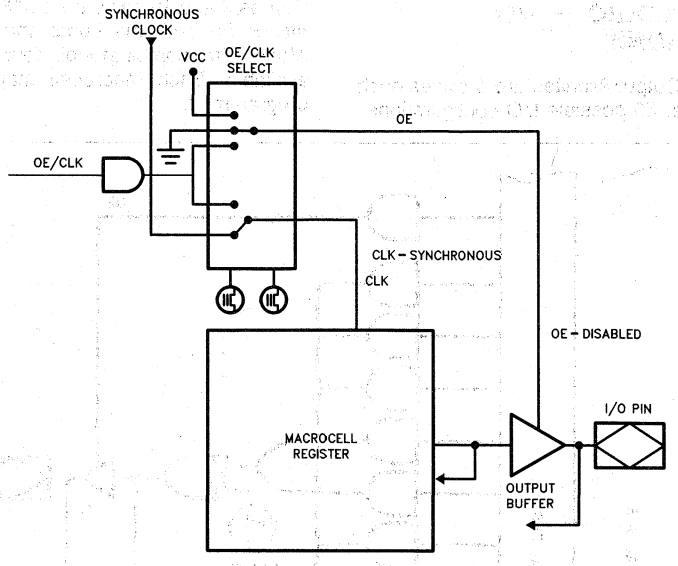
The register is clocked by the quadrant synchronous clock signal which is common to 11 other Macrocells. The output is enabled by the logic from the product term.



290111–13

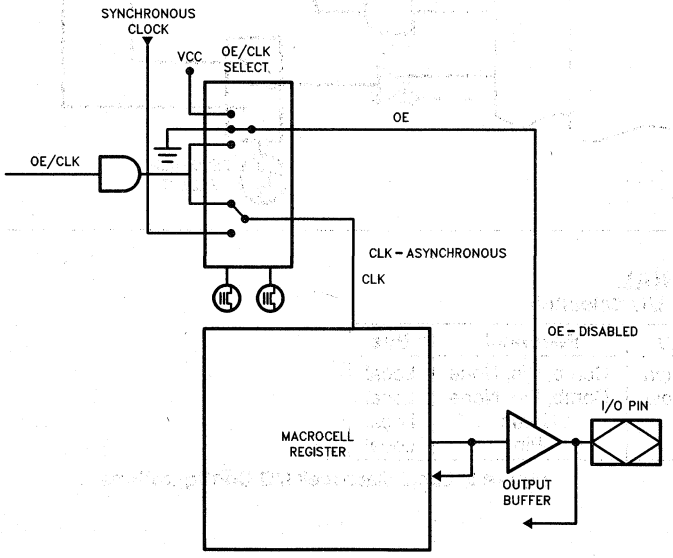
The output is permanently enabled and the register is clocked via the product term. This allows for gated clocks that may be generated from elsewhere in the 5C180.

Figure 8a. Local Macrocell OE/CLK Selection



290111-14

The output is permanently disabled and the register clocked by the quadrant synchronous clock signal. The pin can be used as an input while the register or combinational output can be fed back.



290111-15

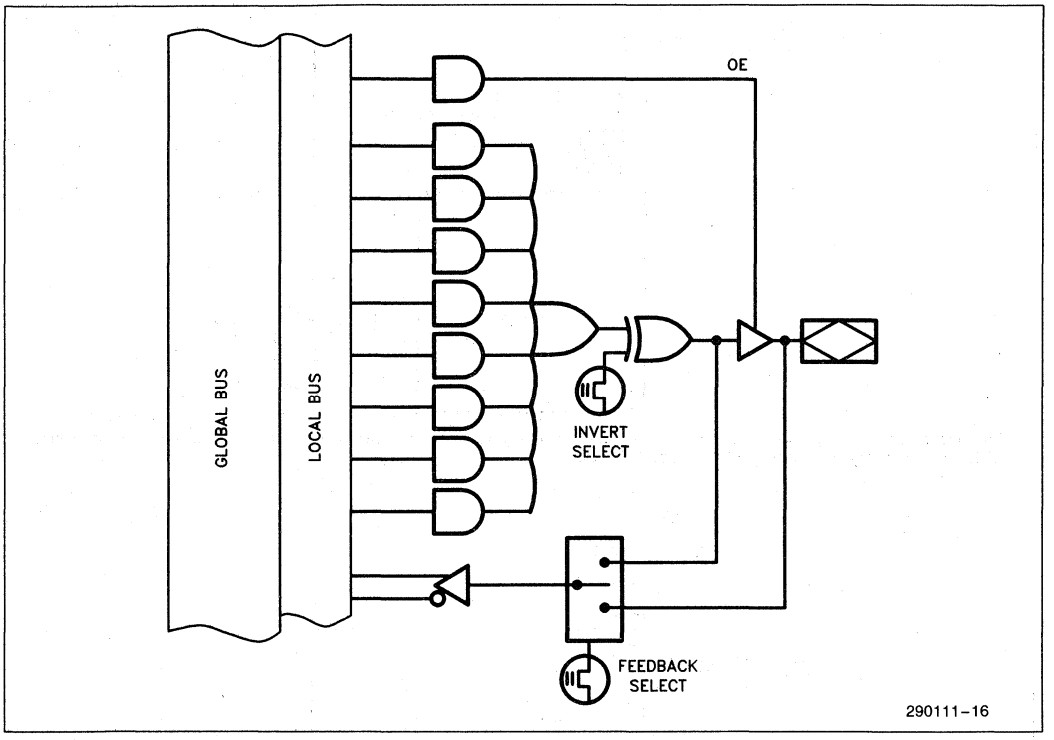
The output is permanently disabled and the register is clocked via the product term. This allows gated clocks that may be generated elsewhere in the 5C180. The pin can be used as in input while the register or combinational output can be fed back.

Figure 8b. Global Macrocell Additional OE/CLK Selection

MACROCELL LOGIC + I/O CONFIGURATIONS

The 5C180 Input/Output Architecture provides each Macrocell with over 50 possible I/O configurations.

Figures 9 and 10 show the 5C180 basic I/O configurations for both the Local and Global Macrocells. Along with combinatorial, four register types are available. Each Macrocell may be independently programmed.

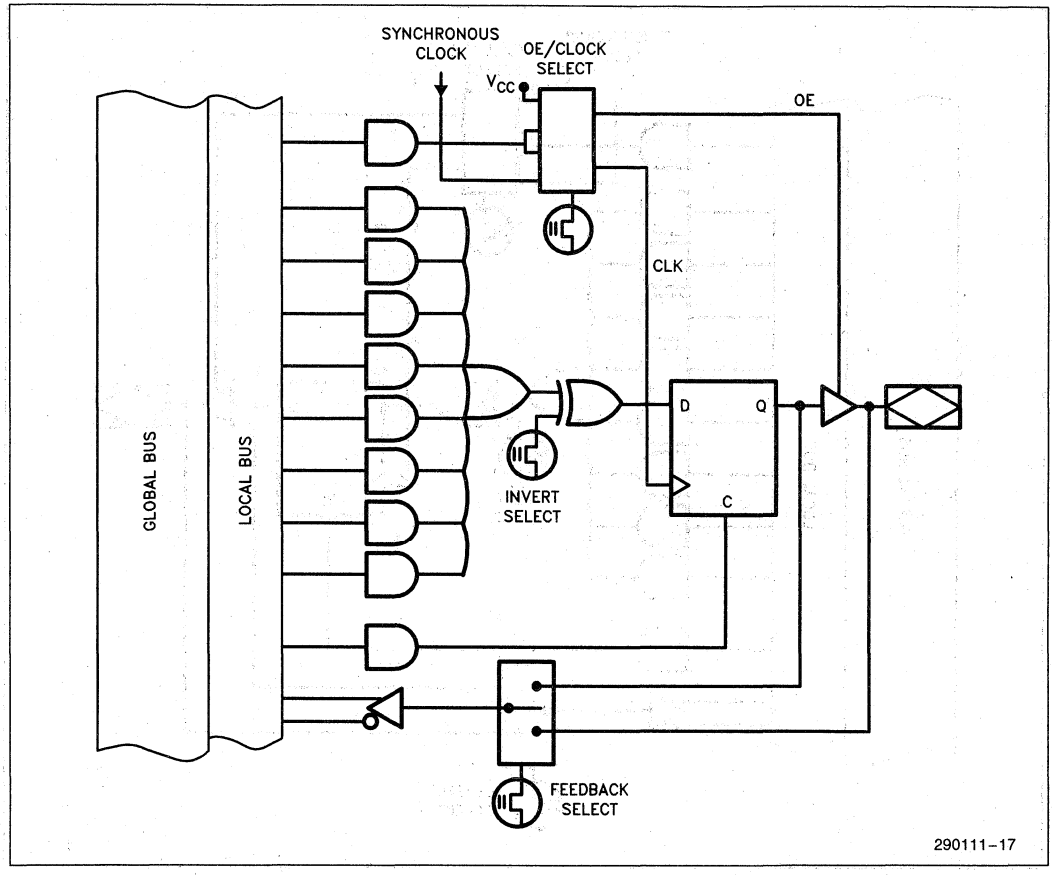


COMBINATORIAL

I/O Selection

Output/Polarity	Feedback	Bus
Combinatorial/High	Comb, Pin, None	Local
Combinatorial/Low	Comb, Pin, None	Local
None	Comb	Local
None	Pin	Local

Figure 9. Local Macrocell I/O Configurations



D-TYPE FLIP-FLOP

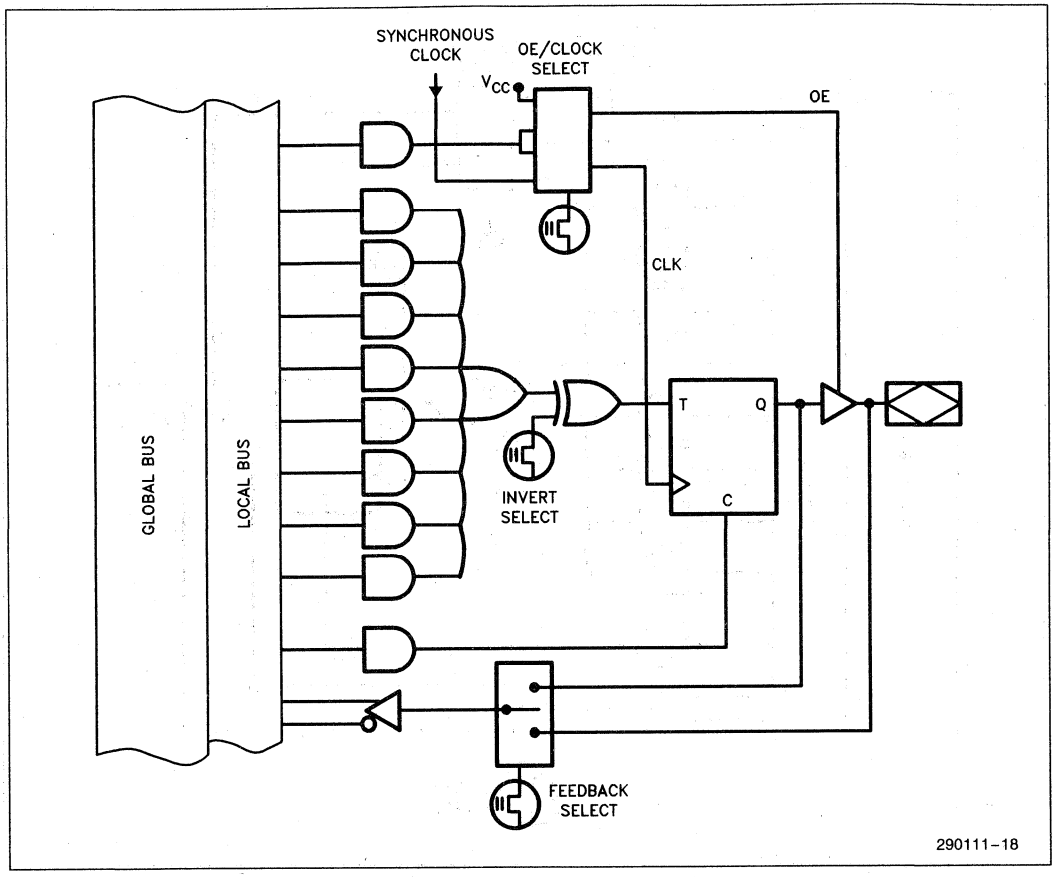
I/O Selection

Output/Polarity	Feedback	Bus
D-Register/High	D-Register, Pin, None	Local
D-Register/Low	D-Register, Pin, None	Local
None	D-Register	Local
None	Pin	Local

Function Table

D	Q_n	Q_{n+1}
0	0	0
0	1	0
1	0	1
1	1	1

Figure 9. Local Macrocell I/O Configurations (Continued)



TOGGLE FLIP-FLOP

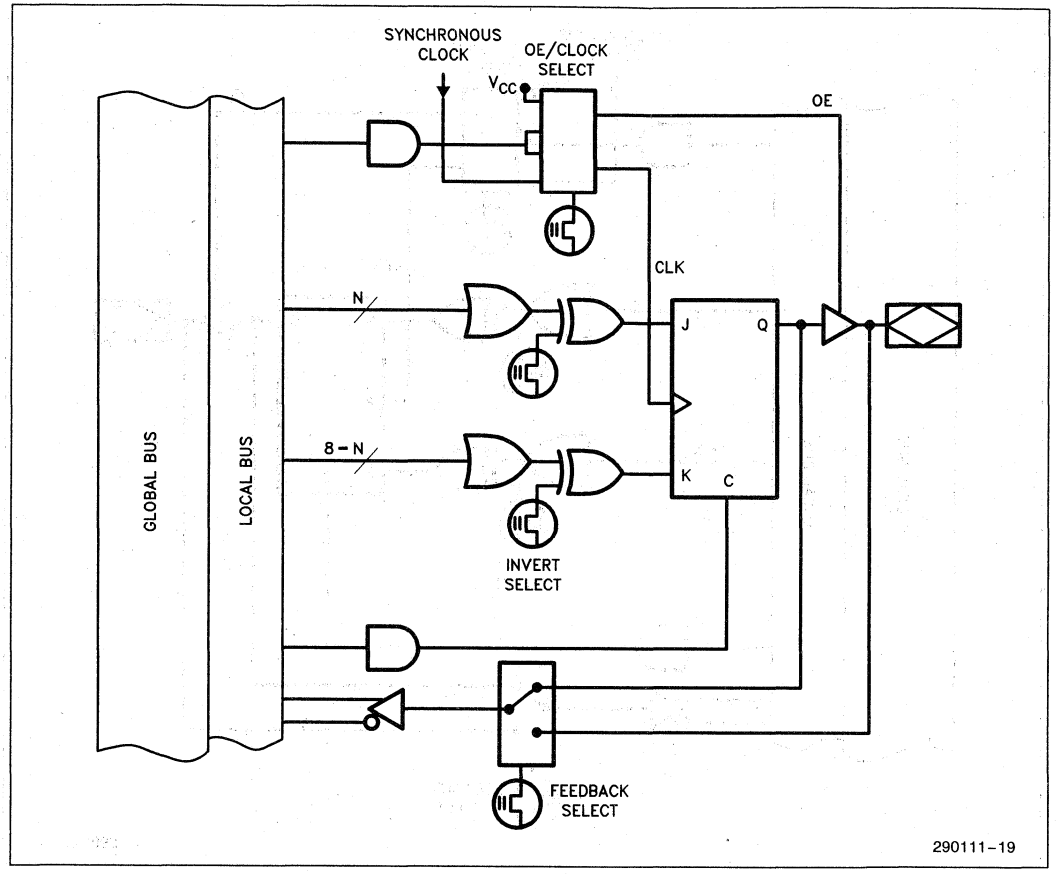
I/O Selection

Output/Polarity	Feedback	Bus
T-Register/High	T-Register, Pin, None	Local
T-Register/Low	T-Register, Pin, None	Local
None	T-Register	Local
None	Pin	Local

Function Table

T	Q_n	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

Figure 9. Local Macrocell I/O Configurations (Continued)



JK FLIP-FLOP

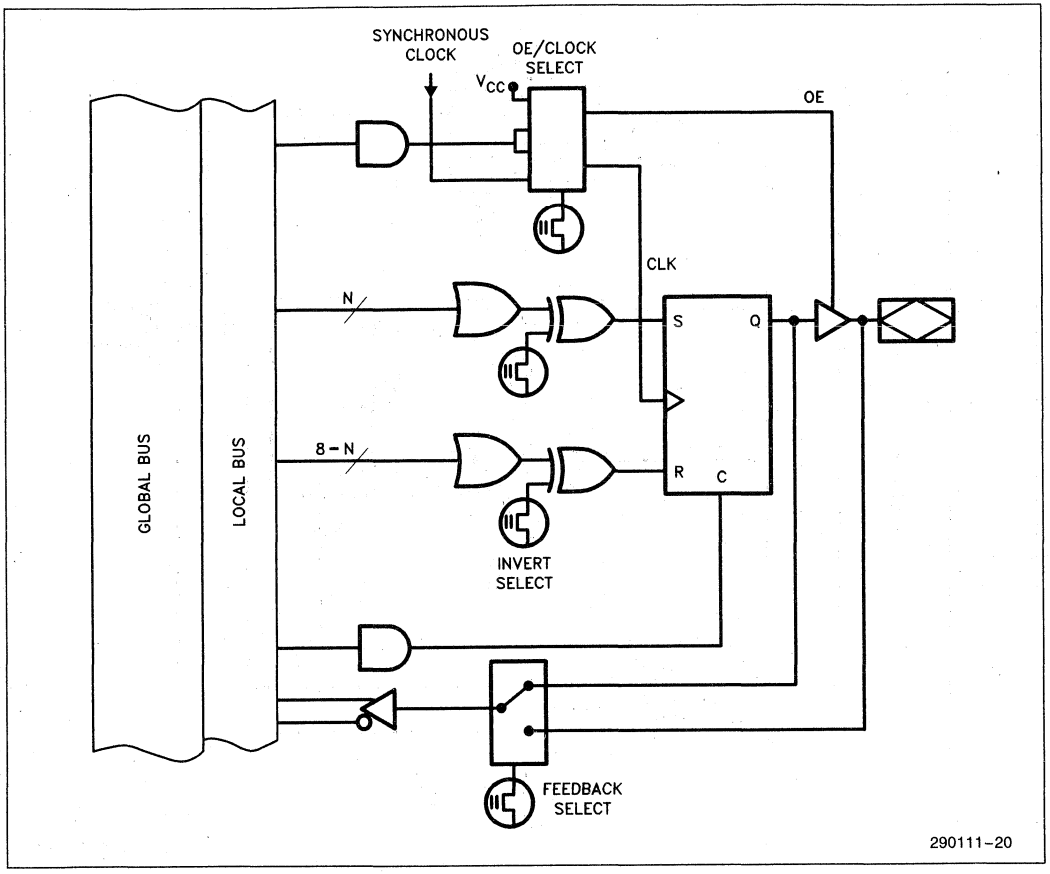
I/O Selection

Output/Polarity	Feedback	Bus
JK Register/High	JK Register, None	Local
JK Register/Low	JK Register, None	Local
None	JK Register	Local

Function Table

J	K	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Figure 9. Local Macrocell I/O Configurations (Continued)



SR FLIP-FLOP

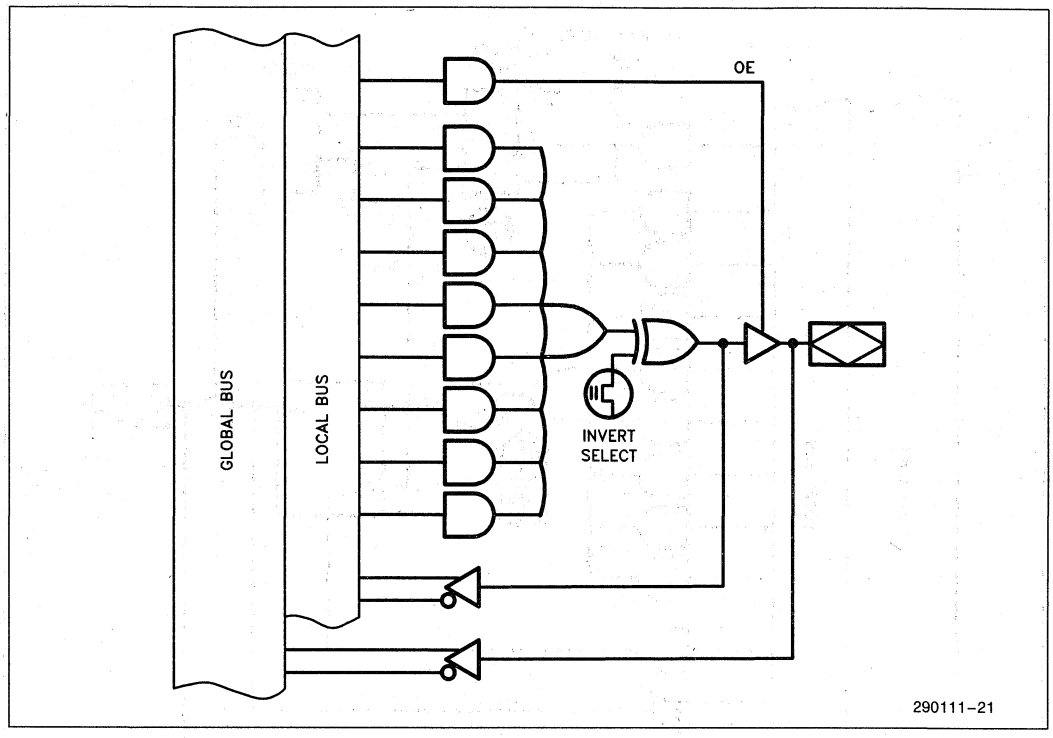
I/O Selection

Output/Polarity	Feedback	Bus
SR Register/High	SR Register, None	Local
SR Register/Low	SR Register, None	Local
None	SR Register	Local

Function Table

S	R	Q_n	$Q_n + 1$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1

Figure 9. Local Macrocell I/O Configurations (Continued)

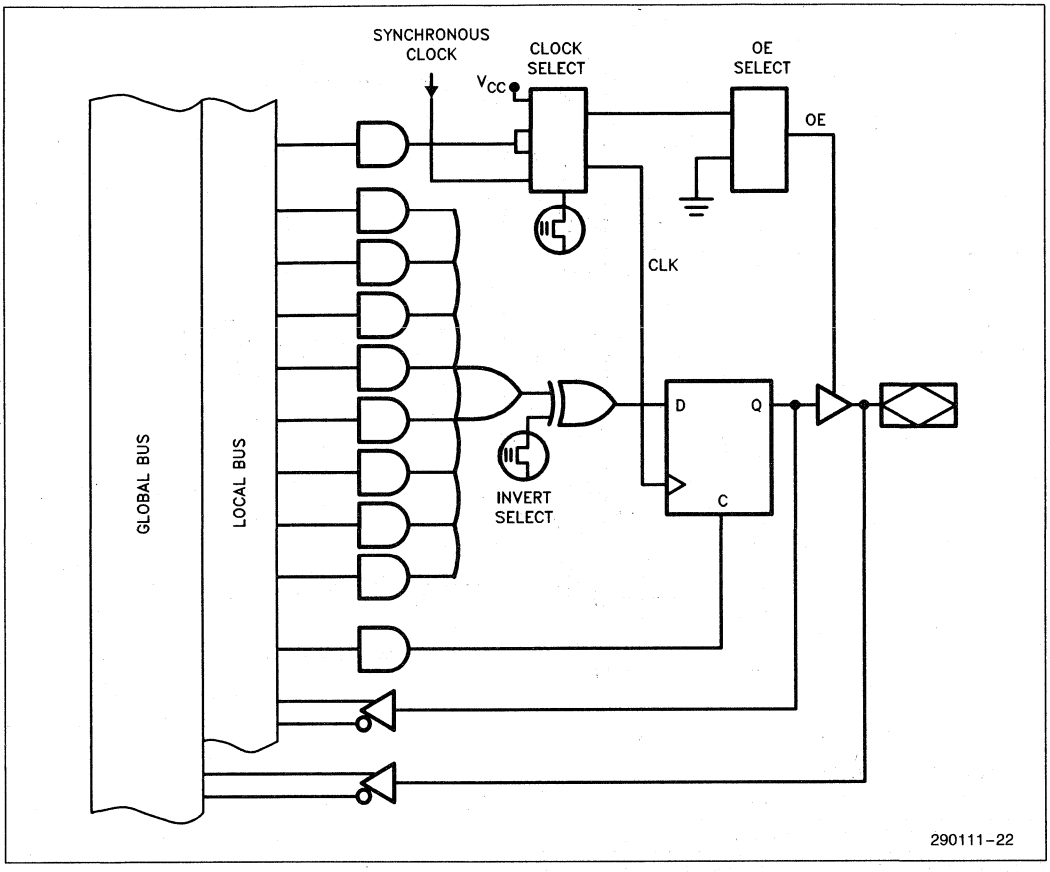


COMBINATORIAL

I/O Selection

Output/Polarity	Feedback	Bus
Combinatorial/High	Comb, Pin, None	Local, Global
Combinatorial/Low	Comb, Pin, None	Local, Global
None	Comb	Local, Global
None	Pin	Global
None	Comb/Pin	Local/Global

Figure 10. Global Macrocell I/O Configurations



D-TYPE FLIP-FLOP

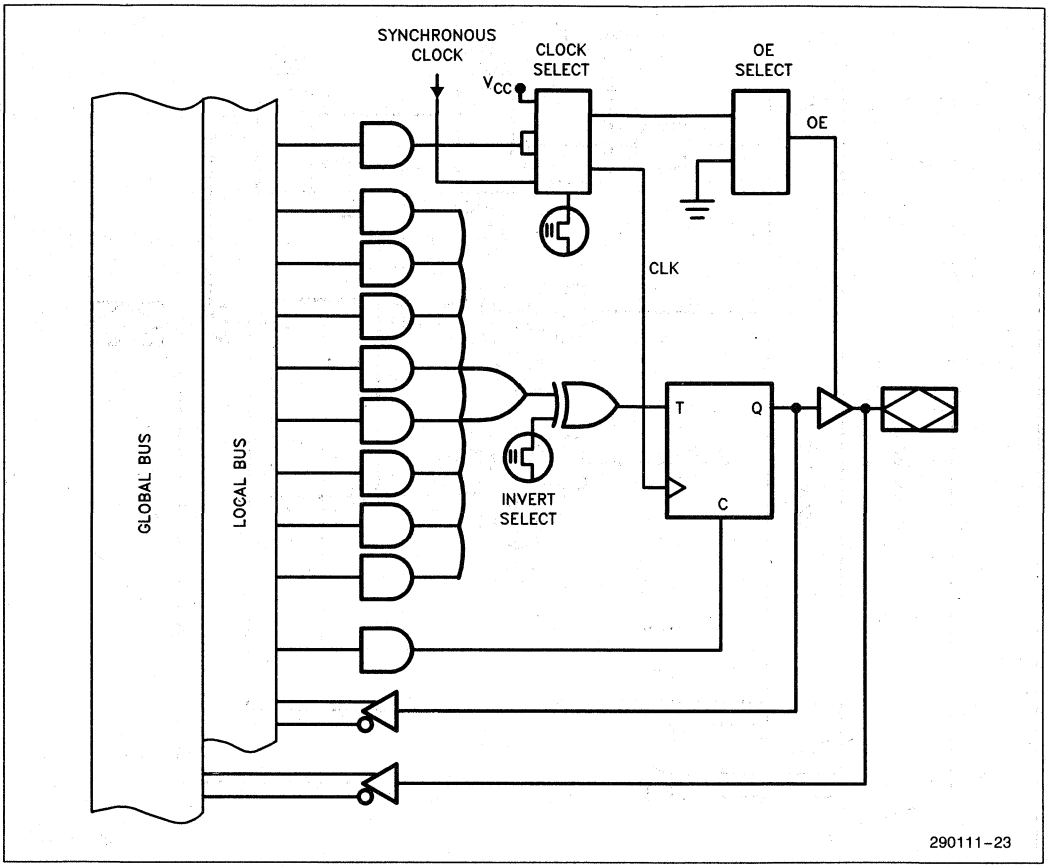
I/O Selection

Output/Polarity	Feedback	Bus
D-Register/High	D-Register, Pin, None	Local, Global
D-Register/Low	D-Register, Pin, None	Local, Global
None	D-Register	Local, Global
None	Pin	Global
None	D-Register/Pin	Local/Global

Function Table

D	Q_n	Q_{n+1}
0	0	0
0	1	0
1	0	1
1	1	1

Figure 10. Global Macrocell I/O Configurations (Continued)



TOGGLE FLIP-FLOP

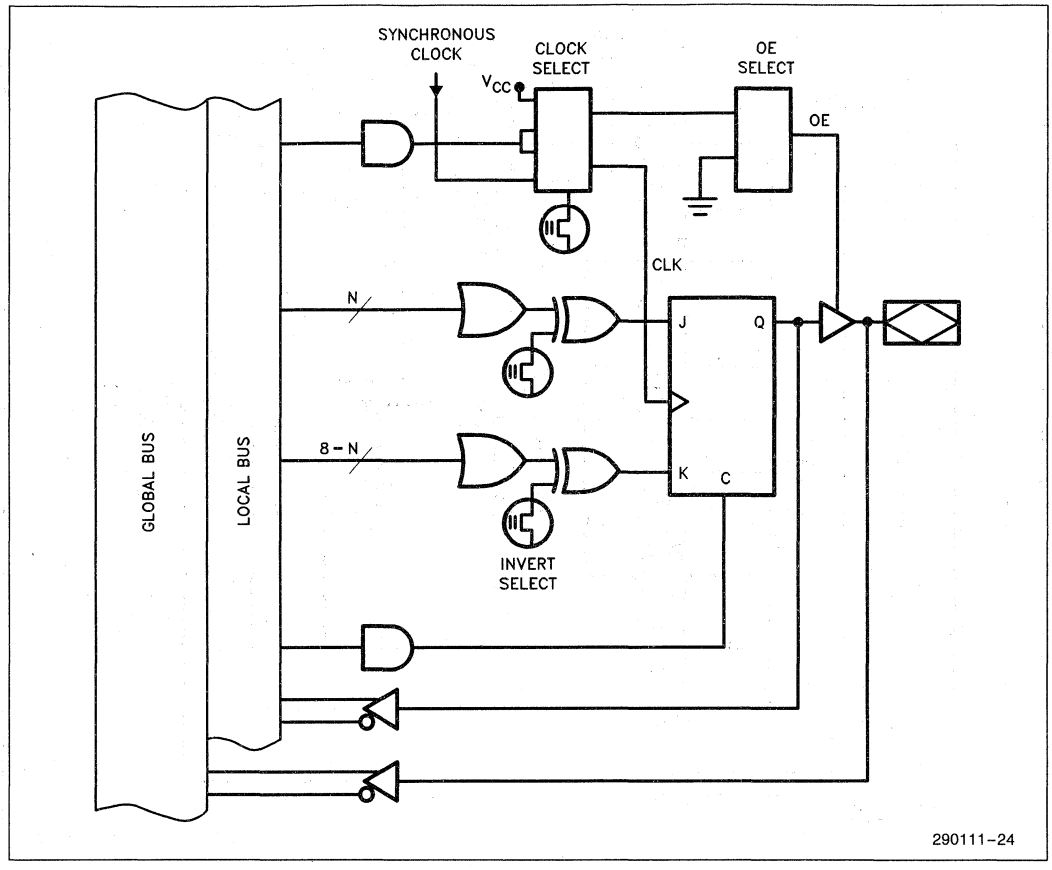
I/O Selection

Output/Polarity	Feedback	Bus
T-Register/High	T-Register, Pin, None	Local, Global
T-Register/Low	T-Register, Pin, None	Local, Global
None	T-Register	Local, Global
None	Pin	Global
None	T-Register/Pin	Local/Global

Function Table

T	Q_n	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

Figure 10. Global Macrocell I/O Configurations (Continued)



JK FLIP-FLOP

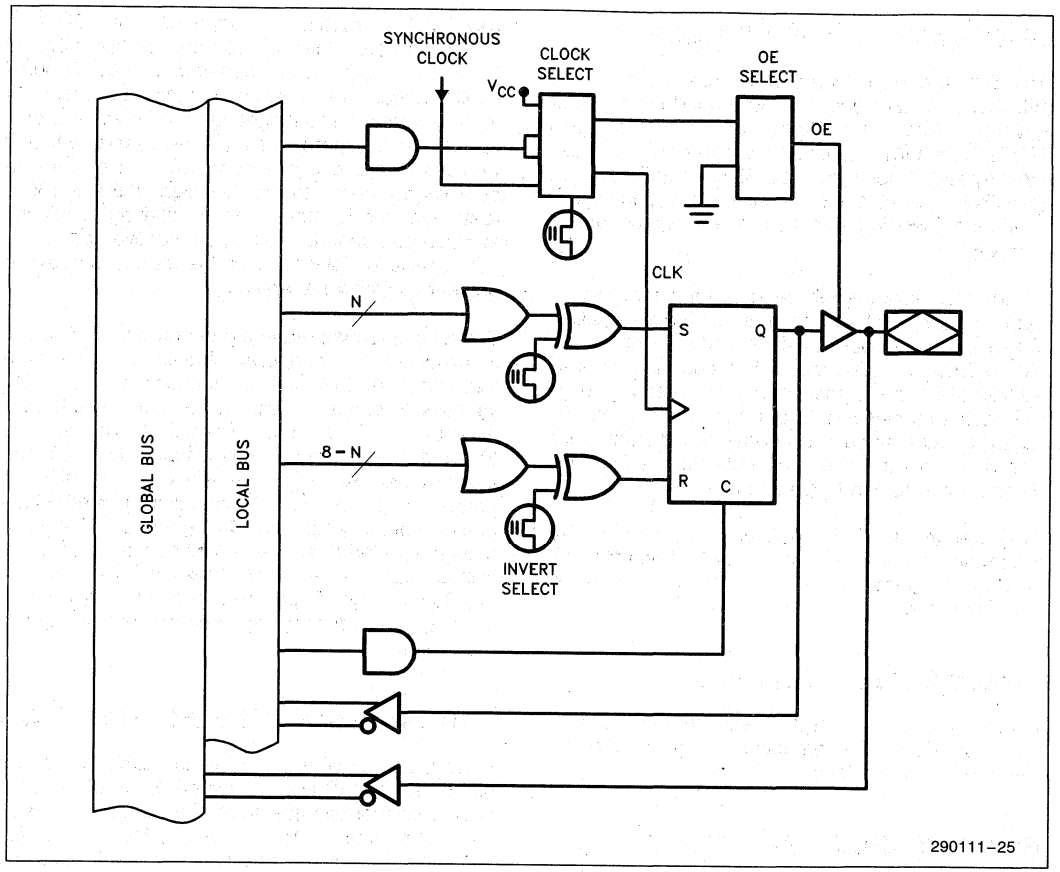
I/O Selection

Output/Polarity	Feedback	Bus
JK Register/High	JK Register, None	Local, Global
JK Register/Low	JK Register, None	Local, Global
None	JK Register	Local
None	JK Register/Pin	Local/Global

Function Table

J	K	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Figure 10. Global Macrocell I/O Configurations (Continued)



SR FLIP-FLOP

I/O Selection

Output/Polarity	Feedback	Bus
SR Register/High	SR Register, None	Local, Global
SR Register/Low	SR Register, None	Local, Global
None	SR Register	Local
None	SR Register/Pin	Local/Global

Function Table

S	R	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1

Figure 10. Global Macrocell I/O Configurations (Continued)

AUTOMATIC STAND-BY MODE

The 5C180 contains a programmable bit, the Turbo Bit, that optimizes operation for speed or for power savings. When the Turbo Bit is programmed (TURBO = ON), the device is optimized for maximum speed. When the Turbo Bit is not programmed (TURBO = OFF), the device is optimized for power savings by entering standby mode during periods of inactivity.

Figure 11 shows the device entering standby mode approximately 100 ns after the last input transition. When the next input transition is detected, the device returns to active mode. Wakeup time adds an additional 30 ns to the propagation delay through the device as measured from the first input. No delay will occur if an output is dependent on more than one input and the last of the inputs changes after the device has returned to active mode.

After erasure, the Turbo Bit is unprogrammed (OFF); automatic standby mode is enabled. When the Turbo Bit is programmed (ON), the device never enters standby mode.

Erased-State Configuration

Prior to programming or after erasing, the I/O structure is configured for combinatorial active low output with input (pin) feedback.

ERASURE CHARACTERISTICS

Erasure characteristics of the 5C180 are such that erasure begins to occur upon exposure to light with wavelengths shorter than approximately 4000Å. It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000Å–4000Å range. Data shows that constant exposure to room level fluorescent lighting could erase the typical 5C180 in approximately three years, while it would take approximately one week to cause erasure when exposed to direct sunlight. If the 5C180 is to be exposed to these types of lighting conditions for extended periods of time, conductive opaque labels should be placed over the device window to prevent unintentional erasure.

The recommended erasure procedure for the 5C180 is exposure to shortwave ultraviolet light with a wavelength of 2537Å. The integrated dose (i.e., UV intensity × exposure time) for erasure should be a minimum of fifteen (15) Wsec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12,000 μW/cm^2 power rating. The 5C180 should be placed within one inch of the lamp tubes during erasure. The maximum integrated dose the 5C180 can be exposed to without damage is 7258 Wsec/cm^2 (1 week at 12,000 μW/cm^2). Exposure to high intensity UV light for longer periods may cause permanent damage to the device.

PROGRAMMING CHARACTERISTICS

Initially, and after erasure, all the EPROM control bits of the 5C180 are connected. Each of the connected control bits are selectively disconnected by programming the EPROM cells into their "on" state. Programming voltage and waveform specifications are available by request from Intel to support programming of the 5C180.

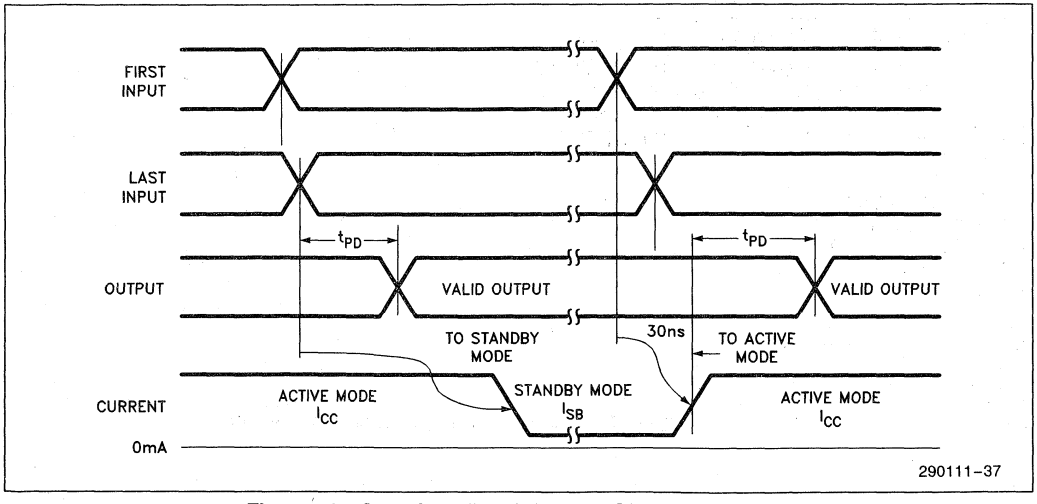


Figure 11. 5C180 Standby and Active Mode Transitions

inteligent Programming™ Algorithm

The 5C180 supports the inteligent Programming Algorithm which rapidly programs Intel PLDs using an efficient and reliable method. The inteligent Programming Algorithm is particularly suited to the production programming environment. This method ensures reliability as the incremental program margin of each bit is continually monitored to determine when the bit has been successfully programmed.

FUNCTIONAL TESTING

Since the logical operation of the 5C180 is controlled by EPROM elements, the device is completely testable. Each programmable EPROM bit controlling the internal logic is tested using application-independent test program patterns. After testing, the devices are erased before shipment to customers. No post-programming tests of the EPROM array are required.

The testability and reliability of EPROM-based programmable logic devices is an important feature over similar devices based on fuse technology. Fuse-based programmable logic devices require a use to perform post-programming tests to insure proper programming. These tests must be done at the device level because of the cummulative error effect. For example, a board containing ten devices each possessing a 2% device fallout translates into an 18% fallout at the board level (it should be noted that programming fallout of fuse-based programmable logic devices is typically 2% or higher).

DESIGN RECOMMENDATIONS

For proper operation, it is recommended that all input and output pins be constrained to the voltage range GND $<$ (V_{IN} or V_{OUT}) $<$ V_{CC}. Unused inputs and I/Os should be tied to V_{CC} or GND to minimize device power consumption. Reserved pins (as indicated in the logic compiler REPORT file) should be left floating (no connect) so that the pin can attain the appropriate logic level. A power supply decoupling capacitor of at least 0.2 μf must be connected directly between V_{CC} and GND.

As with all CMOS devices, ESD handling procedures should be used with this device to prevent damage during programming, assembly, and test.

DESIGN SECURITY

A single EPROM bit provides a programmable design security feature that controls the access to the data programmed into the device. If this bit is set, a proprietary design within the device cannot be copied. This EPROM security bit enables a higher degree of design security than fused-based devices since programmed data within EPROM cells is invisible even to microscopic evaluation. The EPROM security bit, along with all the other EPROM control bits, will be reset by erasing the device.

LATCH-UP IMMUNITY

All of the input, I/O, and clock pins of the 5C180 have been designed to resist latch-up which is inherent in inferior CMOS structures. The 5C180 is designed with Intel's proprietary CHMOS II-E EPROM process. Thus, each of the 5C180 pins will not experience latch-up with currents up to $\pm$100 mA and voltages ranging from $-$1V to (V_{CC} $+$1V). Furthermore, the programming pin is designed to resist latch-up to the 13.5V maximum device limit.

INTEL PROGRAMMABLE LOGIC DEVELOPMENT SYSTEM II (iPLDS II)

iPLDS II provides all the tools needed to design with Intel EPLDs or compatible devices. In addition to providing development assistance, iPLDS II insulates the user from having to know all the intricate details of EPLD architecture (the machine will optimize a design to benefit from architectural features). It contains comprehensive third generation software that supports several different design entry methods, minimizes logic, does automatic pin assignments and produces the best design fit for the selected EPLD. It is user friendly with guided menus, on-line Help messages and soft key inputs.

In addition, iPLDS II contains programmer hardware in the form of an iUP-PC Universal Programmer-Personal Computer to enable the user to program EPLDs, read and verify programmed devices and also to graphically edit programming files. The software generates industry standard JEDEC object code output files which can be downloaded to other programmers as well.

iPLS II software interfaces to several schematic capture packages to enable designs to be entered in schematic form. iPLDview-286/iPLDdraw allows the designer to use familiar TTL symbols or EPLD design primitive symbols. User-defined symbols are also supported. IPLDdraw also provides a path to a.c. timing simulation of EPLD designs.

SCHEMA III-PLD allows the designer to use TTL symbols, EPLD custom macros, or EPLD design primitive symbols. It also supports user-defined symbols.

Other design formats include Boolean equation entry (supported directly by iPLS II) and state machine entry (supported by iSTATE).

For additional information on iPLDS II, refer to the iPLDS II Data Sheet, order number: 290134. Refer to the tools section of the *Programmable Logic* handbook for a complete listing of development tools.

The 5C180 is also supported by third-party logic compilers such as ABEL*, CUPL*, PLDesigner*, Log/IC, etc. Programming support is provided by third-party programmer companies such as Data I/O, Logic Devices, STAG, etc. Please refer to the "Third-Party Support" lists in the *Programmable Logic* handbook for complete information and vendor contacts.

ADF PRIMITIVES SUPPORTED

The following ADF primitives are supported by this device:

INP	JOJF
CONF	JONF
COCF	SONF
COIF	SOSF
RONF	TOIF
RORF	TONF
ROIF	TOTF
NOCF	CLKB
NORF	
NOJF	
NOSF	
NOTF	

ORDERING INFORMATION

t_{PD} (ns)	t_{CO} (ns)	f_{MAX} (MHz)	Order Code	Package	Operating Range
70	29	20.8	N5C180-70	PLCC	Commercial
75	30	19.6	N5C180-75	PLCC	Commercial
90	35	16.1	N5C180-90	PLCC	Commercial

*ABEL is a trademark of Data I/O Corp. CUPL is a trademark of Logical Devices, Inc. PLDesigner is a trademark of MINC, Inc.

ABSOLUTE MAXIMUM RATINGS*

Symbol	Parameter	Min	Max	Units
V_{CC}	Supply Voltage(1)	−2.0	7.0	V
V_{PP}	Programming Supply Voltage(1)	−2.0	13.5	V
V_I	DC Input Voltage(1)(2)	−0.5	$V_{CC}+0.5$	V
t_{stg}	Storage Temperature	−65	+150	°C
t_{amb}	Ambient Temperature(3)	−10	+85	°C

NOTICE: This is a production data sheet. The specifications are subject to change without notice.

**WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

NOTES:
1. Voltages with respect to ground.
2. Minimum DC input is −0.5V. During transitions, the inputs may undershoot to −2.0V or overshoot to +7.0V for periods less than 20 ns under no load conditions.
3. Under bias. Extended temperature versions are also available.

RECOMMENDED OPERATING CONDITIONS

Symbol	Parameter	Min	Max	Units
V_{CC}	Supply Voltage	4.75	5.25	V
V_{IN}	Input Voltage	0	V_{CC}	V
V_O	Output Voltage	0	V_{CC}	V
T_A	Operating Temperature	0	+70	°C
t_R(4)	Input Rise Time		500	ns
t_F(4)	Input Fall Time		500	ns

NOTE:
4. t_R and t_F for clocks is 250 ns.

D.C. CHARACTERISTICS $T_A = 0°$ to $+70°C$, $V_{CC} = 5V \pm 5\%$

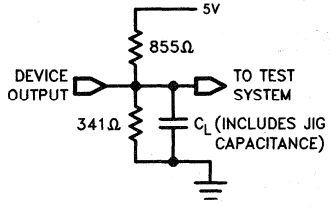
Symbol	Parameter/Test Conditions	Min	Typ	Max	Unit
V_{IH}(5)	High Level Input Voltage	2.0		V_{CC} + 0.3	V
V_{IL}(5)	Low Level Input Voltage	−0.3		0.8	V
V_{OH}(6)	High Level Output Voltage $I_O = -4.0$ mA D.C., V_{CC} = min.	2.4			V
V_{OL}	Low Level Output Voltage $I_O = 4.0$ mA D.C., V_{CC} = min.			0.45	V
I_I	Input Leakage Current V_{CC} = max., GND < V_{IN} < V_{CC}			±10	μA
I_{OZ}	Output Leakage Current V_{CC} = max., GND < V_{OUT} < V_{CC}			±10	μA

NOTES:
5. Absolute values with respect to device GND; all over and undershoots due to system or tester noise are included.
6. I_O at CMOS levels (3.84 V) = −2 mA
7. Not more than 1 output should be tested at a time. Duration of that test must not exceed 1 second.
8. With Turbo Bit Off, device automatically enters standby mode approximately 100 ns after last input transition.

D.C. CHARACTERISTICS $T_A = 0°$ to $+70°C$, $V_{CC} = 5V \pm 5\%$ (Continued)

Symbol	Parameter/Test Conditions	Min	Typ	Max	Unit
I_{SC}(7)	Output Short Circuit Current V_{CC} = max., V_{OUT} = 0.5V		20	30	mA
I_{SB}(8)	Standby Current V_{CC} = max., V_{IN} = V_{CC} or GND, Standby mode		35	150	μA
I_{CC}	Power Supply Current V_{CC} = max., V_{IN} = V_{CC} or GND, No load, Input Freq. = 1 MHz Active mode (Turbo = Off), Device prog. as four 12-bit Ctrs.		30	45	mA

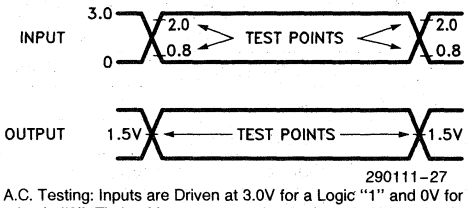
A.C. TESTING LOAD CIRCUIT



290111-28

C_L = 50 pF

A.C. TESTING INPUT, OUTPUT WAVEFORM



290111-27

A.C. Testing: Inputs are Driven at 3.0V for a Logic "1" and 0V for a Logic "0". Timing Measurements are made at 2.0V for a Logic "1" and 0.8V for a Logic "0" on inputs. Outputs are measured at a 1.5V point. Device rise and fall times <6 ns.

CAPACITANCE

Symbol	Parameter	Min	Typ	Max	Unit	Conditions
C_{IN}	Input Capacitance			15	pF	V_{IN} = 0V, f = 1.0 MHz
C_{OUT}	Output Capacitance			15	pF	V_{OUT} = 0V, f = 1.0 MHz
C_{CLK}	Clock Pin Capacitance			25	pF	V_{OUT} = 0V, f = 1.0 MHz
C_{VPP}	V_{PP} Pin Capacitance			160	pF	CLK2, V_{OUT} = 0V, f = 1.0 MHz

A.C. CHARACTERISTICS T_A = 0°C to +70°C, V_{CC} = 5V ±5%, Turbo Bit On(9)

Symbol	From	To	5C180-70 EP1800-2			5C180-75			5C180-90 EP1800			Non-Turbo Mode(11)	Unit
			Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
t_{PD1}	Input(12)	Comb. Output			65			70			85	+30	ns
t_{PD2}	I/O(12)	Comb. Output			70			75			90	+30	ns
t_{PD2e}	I/O(13)	Comb. Output			65			70			85	+30	ns
t_{PZX}(10)	I or I/O	Output Enable			70			75			90	+30	ns
t_{PXZ}(10)	I or I/O	Output Disable			70			75			90	+30	ns
t_{CLR}	Asynch. Reset	Q Reset			70			75			90	+30	ns

NOTES:
9. Typ. Values are at T_A = 25°C, V_{CC} = 5V, Active Mode.
10. t_{PZX} and t_{PXZ} are measured at ±0.5V from steady state voltage as driven by spec. output load. t_{PXZ} is measured with C_L = 5 pF.
11. If device is operated with Turbo Bit Off (Non-Turbo Mode) and the device has been inactive for approx. 100 ns, increase time by amount shown.

SYNCHRONOUS CLOCK MODE A.C. CHARACTERISTICS

T_A = 0°C to +70°C, V_{CC} = 5V ±5%, Turbo Bit On(9)

Symbol	Symbol	5C180-70 EP1800-2			5C180-75			5C180-90 EP1800			Non-Turbo Mode(11)	Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
f_{MAX}	Max Frequency $1/(t_{CH} + t_{CL})$–No Feedback			20.8			19.6			16.1		MHz
f_{CNT}	Max. Count Frequency $1/t_{CNT}$–With Feedback			16.1			15.1			12.2		MHz
t_{SU1}	Input Setup Time to Clk(12)	48			51			62			+30	ns
t_{SU2}	I/O Setup Time to Clk(12)	53			56			67			+30	ns
t_{SU2e}	I/O Setup Time to Clk(13)	48			51			62			+30	ns
t_H	I or I/O Hold after Clk High	0			0			0				ns
t_{CO}	Clk High to Output Valid			29			30			35		ns
t_{CNT}	Register Output Feedback to Register Input–Internal Path	62			66			82			+30	ns
t_{CH}	Clk High Time	24			25			30				ns
t_{CL}	Clk Low Time	24			25			30				ns

ASYNCHRONOUS CLOCK MODE A.C. CHARACTERISTICS

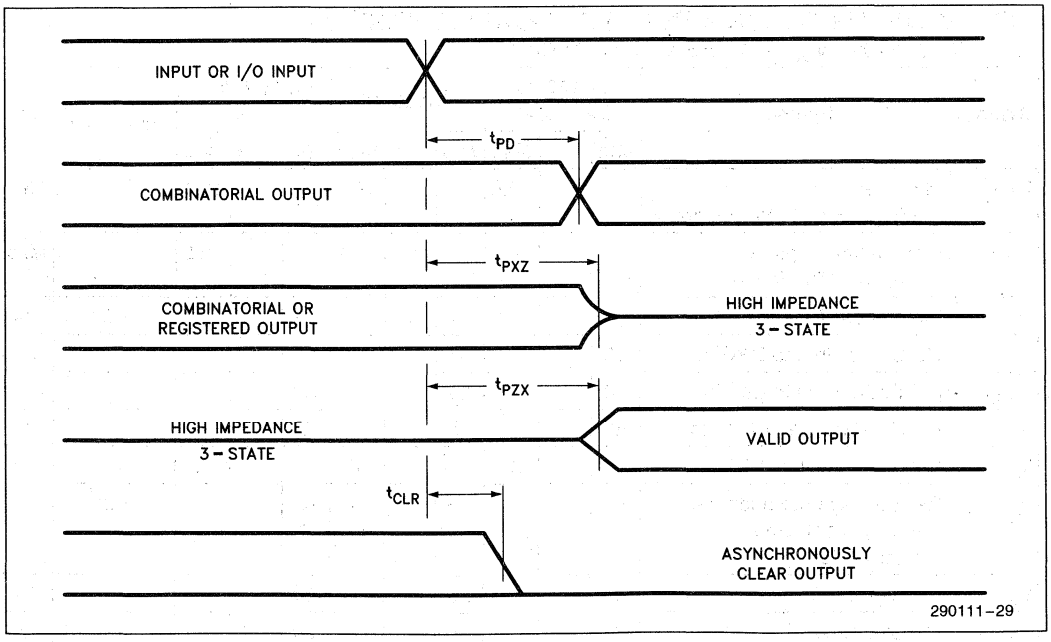
T_A = 0°C to +70°C, V_{CC} = 5V ±5%, Turbo Bit On[9]

Symbol	Parameter	5C180-70 EP1800-2			5C180-75			5C180-90 EP1800			Non-Turbo Mode[11]	Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max		
f_{AMAX}	Max. Frequency $1/(t_{ACH} + t_{ACL})$–No Feedback			20.8			20			16.6		MHz
f_{ACNT}	Max. Frequency $1/t_{ACNT}$–With Feedback			16.1			15.1			12.2		MHz
t_{ASU1}	Input Setup Time to Asynch. Clock[12]	17			19			23			+30	ns
t_{ASU2}	I/O Setup Time to Asynch. Clock[12]	22			25			28			+30	ns
t_{AH}	Input or I/O Hold to Asynch. Clock	30			30			30				ns
t_{ACO}	Asynch. Clk to Output Valid			70			75			90		ns
t_{ACNT}	Register Output Feedback to Register Input–Internal Path	62			66			82			+30	ns
t_{ACH}	Asynch. Clk High Time	24			25			30				ns
t_{ACL}	Asynch. Clk Low Time	24			25			30				ns

NOTES:
12. For General and Global Macrocells.
13. For Enhanced Macrocells.

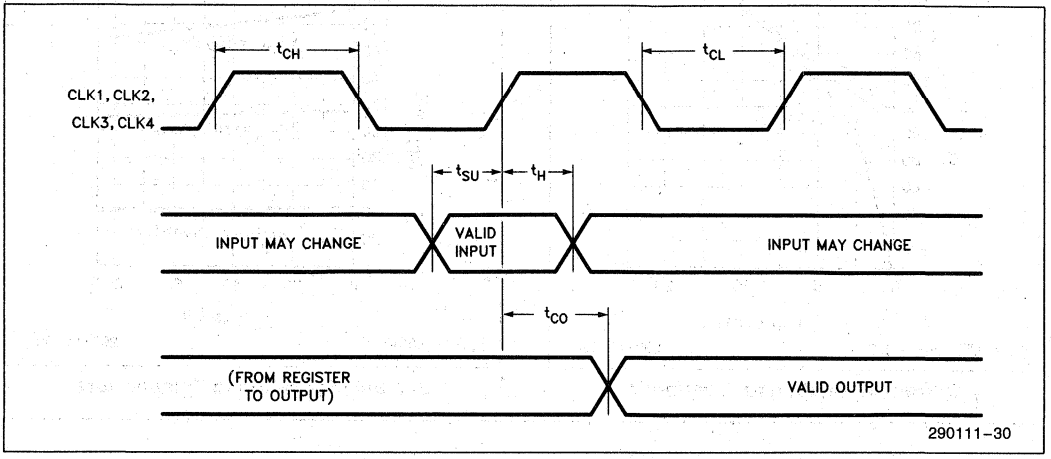
SWITCHING WAVEFORMS

COMBINATORIAL MODE

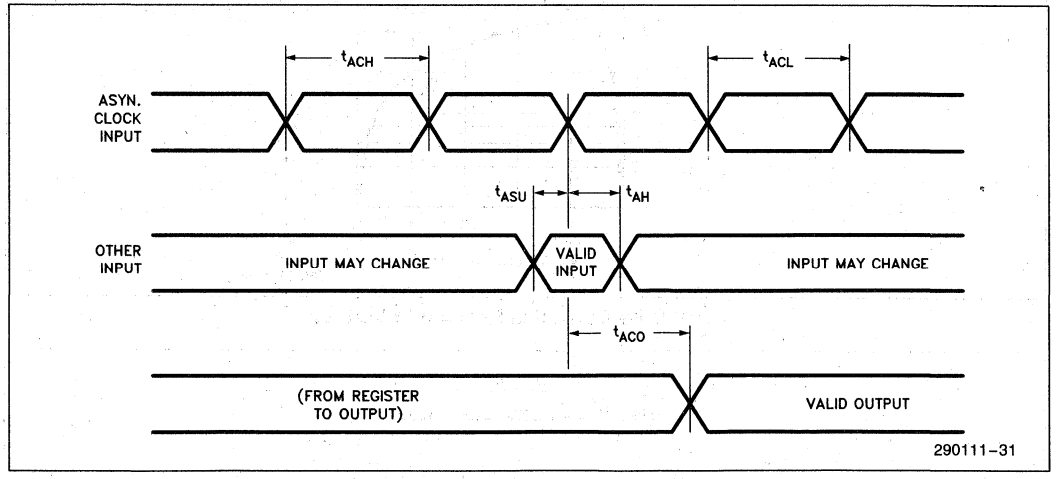


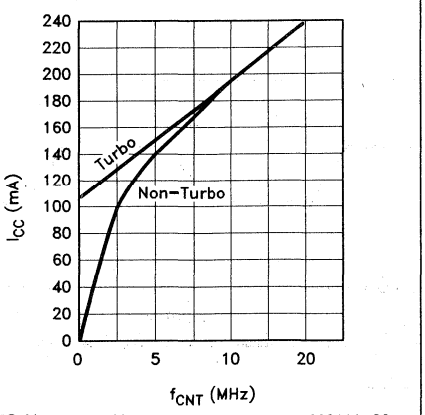
SWITCHING WAVEFORMS (Continued)

SYNCHRONOUS CLOCK MODE



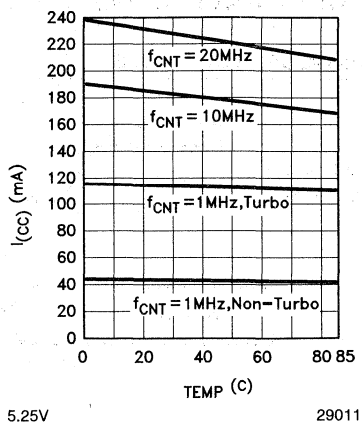
ASYNCHRONOUS CLOCK MODE





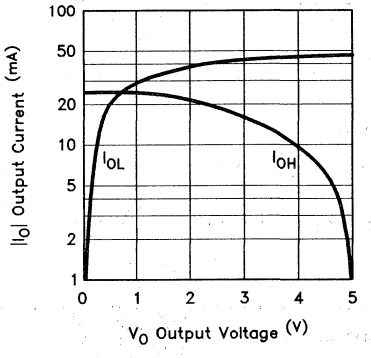
T_A = 0°C, V_{CC} = 5.25V 290111-32

Current in Relation to Frequency



V_{CC} = 5.25V 290111-33

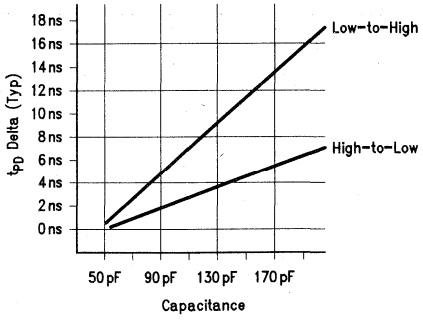
Current in Relation to Temperature



290111-34

Output Drive Current in Relation to Voltage

5C180
t_{PD} Derating vs Capacitive Loading

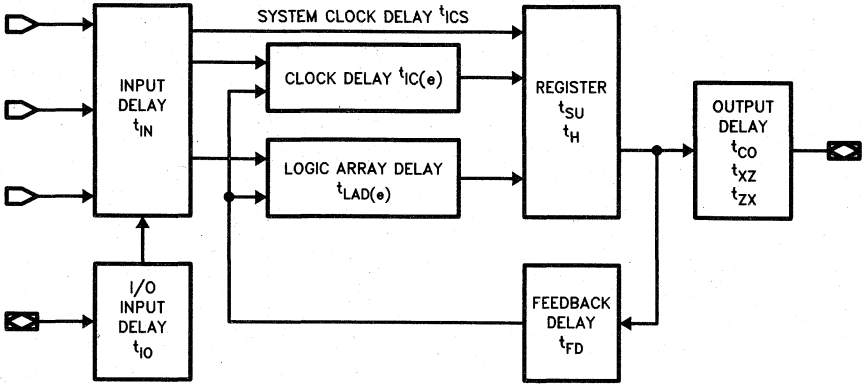


290111-39

Conditions:
T_A = 25°C
V_{CC} = 5.0V

5C180 INTERNAL TIMING

The following internal timing model and specifications are provided to aid in determining the different timing parameters for all permutations of timing paths through the device. The mnemonics in the table represent *internal parameters* only and should not be confused with external timing parameters shown in previous tables, even though some mnemonics are the same.



290111-38

Symbol	Parameter	5C180-70 EP1800-2		5C180-75		5C180-90 EP1800		Non-Turbo Mode(11)		Unit
		Min	Max	Min	Max	Min	Max	Min	Max	
t_{IN}	Input Pad and Buffer Delay		10		11		14		0	ns
t_{IO}	I/O Input Pad and Buffer Delay		5		5		5		0	ns
t_{LADe}	Enhanced Logic Array Delay		35		37		43		30	ns
t_{LAD}	Logic Array Delay		40		42		48		30	ns
t_{OD}	Output Buffer and Pad Delay		15		17		23		0	ns
t_{ZX}	Output Buffer Enable		15		17		23		0	ns
t_{XZ}	Output Buffer Disable		15		17		23		0	ns
t_{SU}	Register Setup Time	12		13		18		0		ns
t_{HS}	Register Hold Time (System Clock)	0		0		0		0		ns
t_H	Register Hold Time	30		30		30		0		ns
t_{ICe}	Enhanced Clock Delay		35		37		43		30	ns
t_{IC}	Clock Delay		40		42		48		30	ns
t_{ICS}	System Clock Delay		4		4		4		0	ns
t_{FD}	Feedback Delay		10		11		16		−30	ns
t_{CLRe}	Enhanced Register Clear Time		35		37		43		30	ns
t_{CLR}	Register Clear Time		40		42		48		30	ns

Applications Information 3

April 1989

High-Speed System Design Using the 85C508 μPLD

DANIEL E. SMITH
PROGRAMMABLE LOGIC APPLICATIONS
INTEL CORPORATION

Order Number: 292055-001

HIGH-SPEED SYSTEM DESIGN USING THE 85C508 μPLD

CONTENTS

INTRODUCTION

Overall performance of microcomputer systems starts with the speed of the system's microprocessor, but goes far beyond simply the processor. System performance also depends on the ability of the memory and I/O subsystems to keep the processor running at full speed. As microprocessors continue to push the speed barrier, pressure is increasingly placed on the rest of the system to keep up. If the subsystems do not increase in speed along with the processor, the resulting system will be unable to take maximum advantage of a processor's potential performance.

The Intel 85C508 Decoder/Latch μPLD is the first member of Intel's μPLD (Microcomputer Programmable Logic Device) family of CHMOS III devices specifically designed for high-speed microcomputer systems. The inherent speed of the 85C508 (available with maximum propagation delays of 7.5, 10, and 15 ns) allows it to be used throughout systems to speed up key interfaces or subsystems. The 85C508 can help your system get the highest possible performance from its microprocessor.

Figure 1 shows the architecture of the 85C508, with its sixteen direct inputs, simplified NAND p-term array,

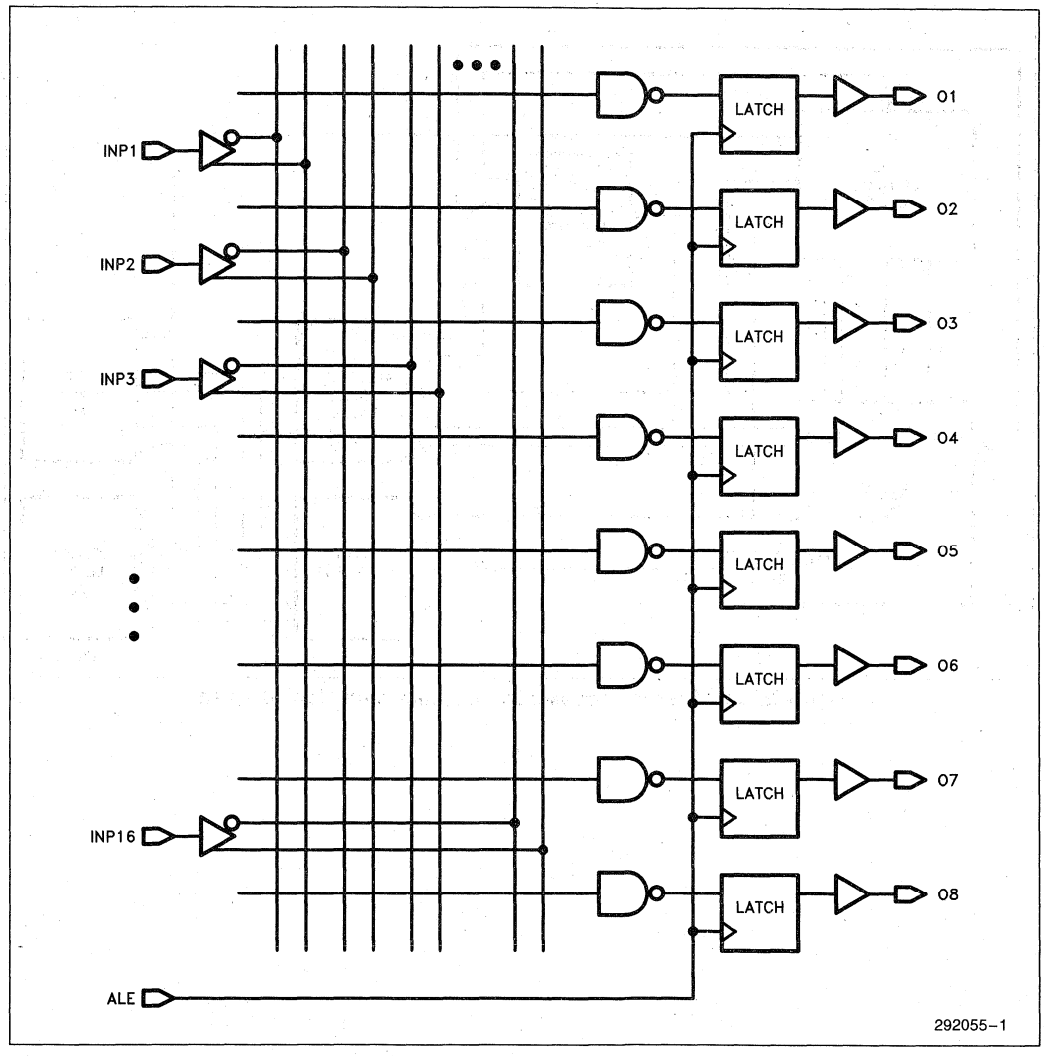


Figure 1. 85C508 Architecture

and eight transparent output latches driven by a global latch enable. The architecture is tailored for speed paths requiring both latching and a degree of decoding or data manipulation. Integration of decode logic and latches increases performance by reducing the propagation delay and setup times between discrete devices. Integration also help conserve precious printed circuit board real estate. CHMOS III technology contributes to enormous power savings over other technologies.

Three different applications of the 85C508 device are shown here: (1) main system address decoding versus alternative approaches, (2) cache memory decoding, and (3) cascading devices for general-purpose combinatorial logic. Information is also provided to aid board design and design entry.

MAIN SYSTEM ADDRESS DECODING

The most straightforward approach to optimizing system speed is simply to use faster devices in the interface between the processor and the subsystems. In a memory interface, for example, faster interface devices allow slower memory devices to be used, thereby reducing memory costs significantly. An alternative is to use the

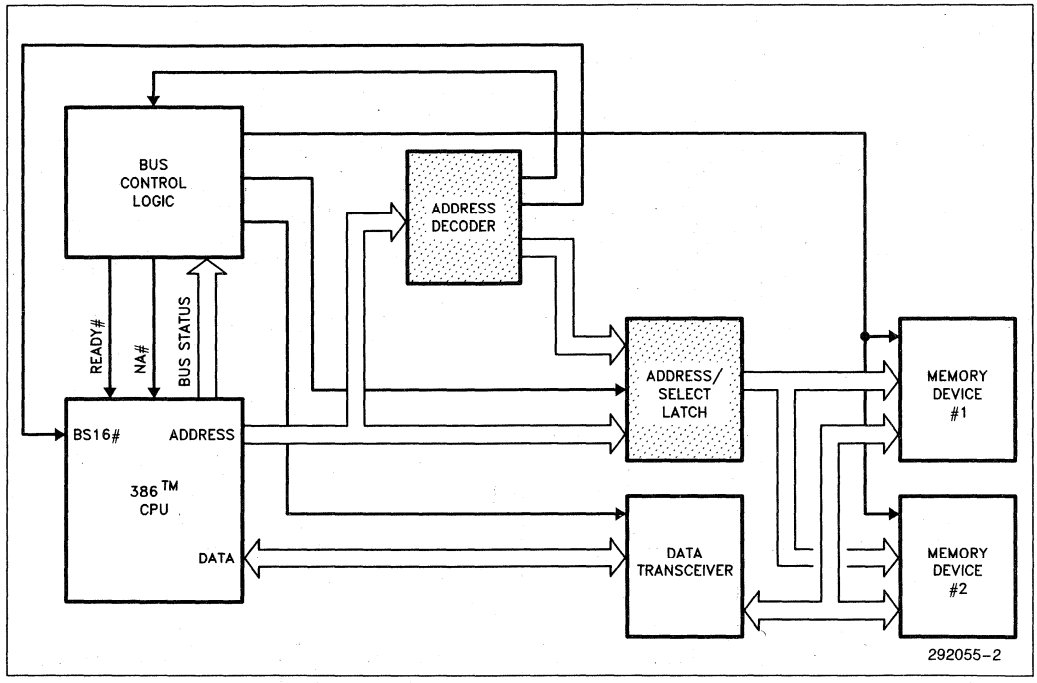


Figure 2. 386TM Microprocessor Memory Interface Block Diagram

faster interface devices together with fast memory to gain additional performance.

The 85C508 provides a high-performance solution to main system address decoding/latching requirements. Figure 2 shows the major paths between a 386TM microprocessor and memory. The 85C508 can speed up the system address decode and system address latch blocks shown in the figure, including the critical paths back to the bus control logic. While the timing in this section is based on the 386 microprocessor, the same principles apply to systems based on other processors.

Non-Pipelined Bus Cycles

The 386 CPU is capable of performing both non-pipelined and pipelined bus cycles, as shown in Figure 3. Address information for non-pipelined cycles is generated at the start of each bus cycle and is held valid until the end of the cycle. Since the address is stable during most of the cycle, and since the 386 CPU has separate address and data buses, straightforward address decoding is all that is required for non-pipelined cycles. Neither the address nor device selects need to be latched.

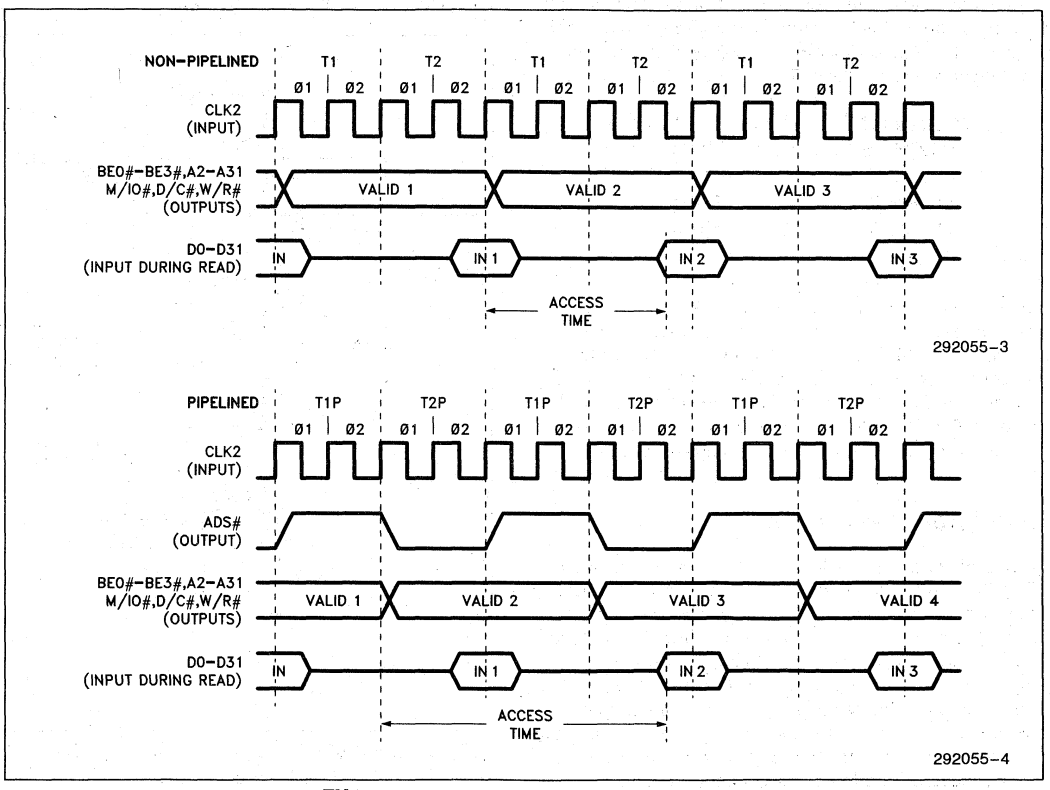


Figure 3. 386TM CPU Non-Pipelined and Pipelined Memory Cycles

386 is a trademark of Intel Corporation.

The tradeoff for this simplicity is a short access time, which puts a speed burden on system memory and I/O.

The 85C508 easily handles non-pipelined decoding. By tying the ALE input high (flow-through latch mode), the device functions as combinatorial logic, which facilitates address decoding for non-pipelined bus cycles. Up to 16 inputs can be decoded to provide up to 8 outputs. For even wider inputs, devices can be cascaded (refer to the discussion on cascading devices later in this application note).

The total access time for 386 CPU non-pipelined bus cycles is shown below. Time is measured from a stable address to data setup (for a read) or setup to end of cycle (for a write). Read access timing is shown.

Processor	Total Access Time
80386-16	78 ns
80386-20	59 ns
80386-25	52 ns
*80386-33	40 ns

*Preliminary information

The times shown include: (1) address decode time, (2) data access time, and (3) any buffer delays. At speeds of 7.5 ns, 10 ns, and 15 ns, the 85C508 provides the fastest address decode time in a CMOS PLD, allowing the widest possible window for data access time and buffer delays. The 16 inputs also allows the 85C508 to decode more signals than is typical with high-speed PLDs.

Pipelined Bus Cycles

For pipelined cycles, the address is made available before the end of the previous cycle and is removed before the end of the current cycle. Because the address is available earlier with respect to the time data is transferred, memory access time is relaxed, allowing the use of slower memory and I/O. With pipelined bus cycles, however, the address and device selects must be latched before the address is removed, requiring additional latches in the system. Thus pipelined cycles trade off a longer access time for increased device count.

Latching of the address and device selects is based on the timing of an ALE (Address Latch Enable) signal. 386 microprocessor-based systems often rely on external logic to generate ALE (the *386 Microprocessor Hardware Reference Manual* shows a PLD generating ALE). This allows the ALE signal to be tailored somewhat to the characteristics of the latches and decoding circuit being used. With this in mind, the following comparison is based not on a fixed ALE timing, but rather on the timing of alternate methods of address decoding. The results can then be applied to design of a PLD-based ALE generation circuit. A faster (and simpler) alternative is also shown.

Figure 4 shows two 20R4 PALs ® used to generate eight device selects from sixteen inputs. As shown, one 85C508 performs the equivalent function. Table 1 compares the timing difference between the two approaches. Note that the equivalent 85C508 speeds can do the same job faster than the two 20R4s operating in parallel. This is due to the high-speed CMOS technology used in the 85C508 and the use of transparent latches on the outputs in place of registers. Device count is also half that of the 20R4 solution.

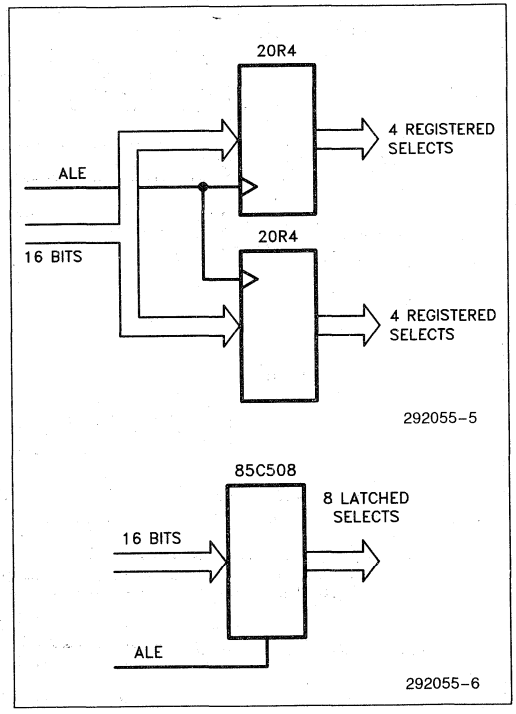


Figure 4. 85C508 Compared with Two 20R4s—Pipelined Bus Cycle

Note that the enable-to-output time shown for the 85C508 implies a setup time to enable. (The implied setup time is the maximum propagation delay minus the enable-to-output time.) The time gained by using the 85C508 is computed by subtracting (1) the sum of the 85C508 setup (implied) and enable-to-output times

PAL is a registered trademark of Monolithic Memories, Inc.

from (2) the sum of the 20R4 setup and clock-to-output times. For the 20R4-7 and 85C508-7 devices, the results are as follows:

	20R4	85C508
Setup	7.5 ns	3.0 ns
Clock/Enable-to-Output	6.5 ns	4.5 ns
Sum	14 ns	7.5 ns
Overall 85C508 Gain	6.5 ns (14 ns − 7.5 ns)	

Figure 5 shows the timing associated with both approaches. Note that 85C508 timing allows ALE to be generated at least 4.5 ns earlier than with the PAL solution. The shorter enable-to-output time cuts an additional 2 ns off the time a device select or control signal could be made available. The potential gain with the 85C508 is 6.5 ns.

One other factor to consider is the flexibility the transparent output latch offers. Even if ALE were to be generated *at the same time* as the address was made available, selects would be reliably decoded. Since the implied setup time would not be met to achieve the enable-to-output time, the selects in this case would not be stable until 7.5 ns after ALE goes high (max. propagation delay).

Table 1. Timing Comparison—20R4 (2x) Vs. 85C508

PAL Device and Series	Setup	Clock To Output	85C508 Speed	Implied Setup	Enable To Output	Gain With 85C508
20R4B Series	15 ns	12 ns	85C508-15	5 ns	10 ns	12 ns
20R4-10 Series	10 ns	8 ns	85C508-10	4 ns	6 ns	8 ns
20R4-7 Series	7.5 ns	6.5 ns	85C508-7	3 ns	4.5 ns	6.5 ns

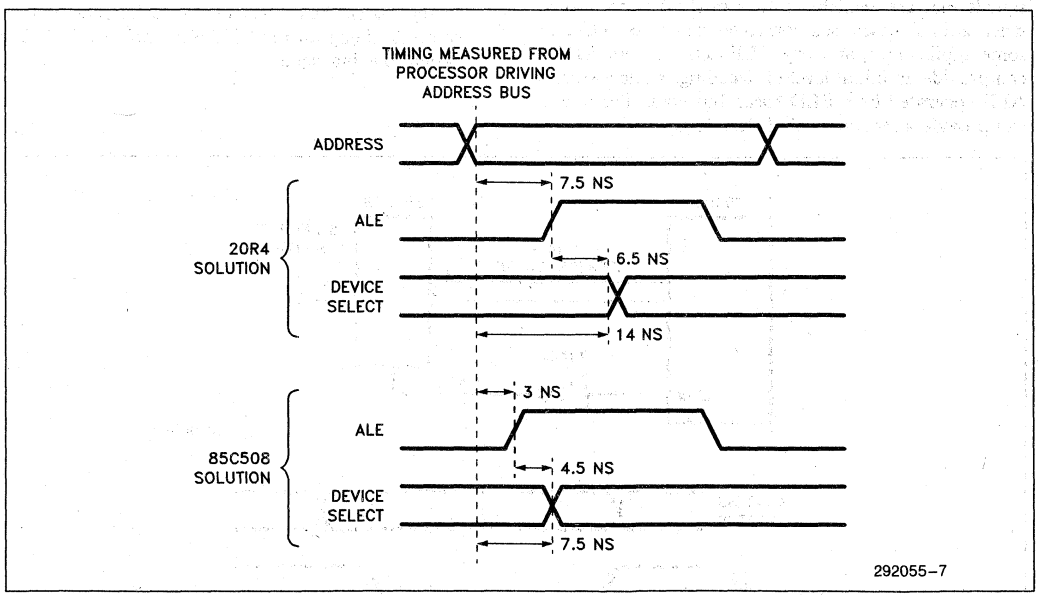


Figure 5. Timing Comparison Between 20R4 and 85C508 ALE Solutions

The same timing comparison applies to a 20R8-7 device. While the 20R8 provides 8 outputs (like the 85C508), these outputs are available only by reducing the number of inputs from 16 to 12. The 85C508, with 16 inputs *and* 8 outputs, is designed for the wider decoding needs of high-performance systems. Close attention to Table 1 and Figure 5 shows that some of the overall gain with the 85C508 is in setup time and some in enable-to-output time. The ALE generation circuit should be designed with both these parameters in mind to achieve the maximum performance benefit provided by the 85C508.

The previous comparison assumes that a PLD-based bus controller generates ALE. Actually, there is a faster approach to address decoding. With the 85C508, the 386 CPU ADS# signal can be used to generate an *early* ALE. The 386 CPU drives ADS# (Address Stable) active (low) as the processor's address pins are driven active. ADS# is driven inactive (high) before the address is removed. Thus ADS# can be routed through an inverter to generate an active high ALE to an 85C508. Figure 6 shows this configuration, along with timing parameters for decoding the address/status signals. An ALE derived from ADS# results in select signals that are available earlier in the bus cycle than when a PLD-based bus controller generates ALE. In some applications, an early ALE derived from ADS# can provide an initial level of decoding, while a second ALE generated by a PLD-based bus controller circuit can provide a second level of decoding.

Figure 6 shows ADS# being generated at the same time as the address bus is being driven active. In reality, ADS# and the address may be skewed somewhat from each other. This prohibits a registered-PAL solution, because PALs require a setup time to ALE. With its transparent latch, however, the 85C508 permits use of ADS# as the source for ALE.

Up to this point, all discussion of decoding has centered around selects to memory and I/O devices. There are additional areas where fast decoding is needed for high-performance 386 microprocessor-based systems. These areas include:

- READY#, NA# (Next Address Request), and BS16# (Bus Size 16) inputs to the 386 CPU (directly or via bus control logic).
- LBA# (Local Bus Access), NCA# (Non-Cacheable Access), and X16# (16-Bit Access) inputs to the 80385 Cache Controller. These signals are typically decoded from the 386 CPU address/status signals.
- BNA# (Bus Next Address) input to the 80385 Cache Controller.
- Since the 80385 provides a 386 CPU-like interface to the system bus, most of the techniques used to speed decoding for the CPU also apply to the 80385 system bus interface.

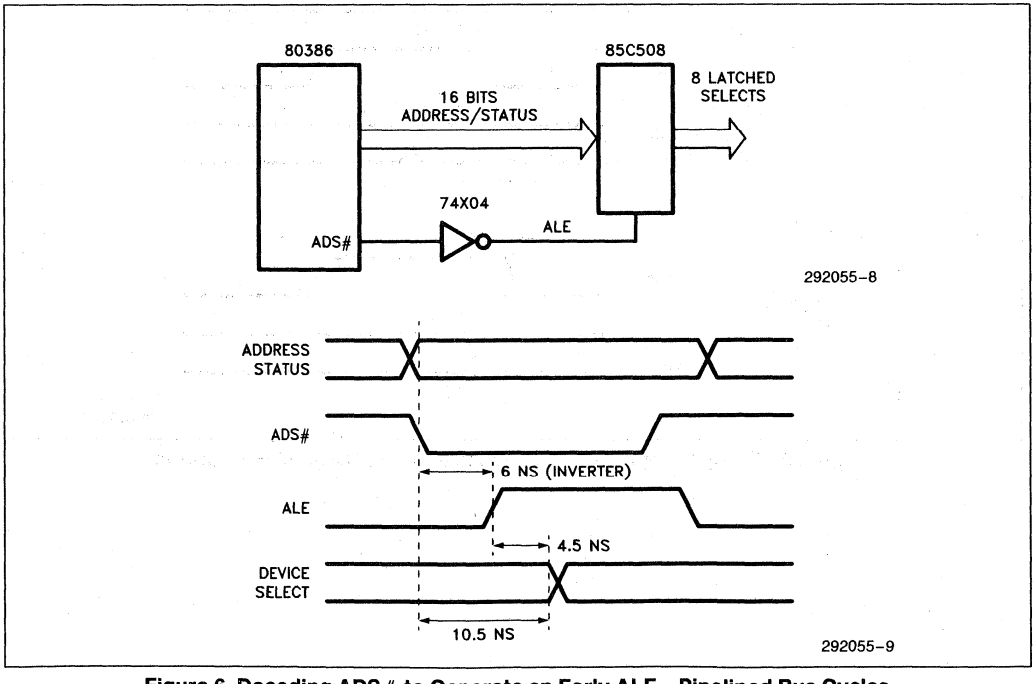


Figure 6. Decoding ADS# to Generate an Early ALE—Pipelined Bus Cycles

Power Consumption

Inherent in its CHMOS process, the 85C508 consumes much less power than its alternatives. Table 2 shows the power consumed by the alternate solutions.

Table 2. Power Consumption of Alternate Solutions

PAL Device and Series	Current	85C508 Speed	Current
20R4B Series (2x)	420 mA	85C508-15	48 mA
20R4-10 Series (2x)	420 mA	85C508-10	48 mA
20R4-7 Series (2x)	420 mA	85C508-7	48 mA
PAL Device and series	**Current**	**85C508 Speed**	**Current**
20R8B Series	210 mA	85C508-15	48 mA
20R8-10 Series	210 mA	85C508-10	48 mA
20R8-7 Series	210 mA	85C508-7	48 mA

CACHE INTERFACE

Cache memory is another approach to increasing overall system performance. The purpose of cache memory is to keep the main processor running as fast as possible by substituting fast SRAM memory cycles for much slower DRAM cycles. By keeping a copy of the most frequently used data in a local cache of fast SRAM, DRAM cycles that require one or more wait states can be greatly reduced, and overall system performance increased.

The 85C508 can provide extremely fast RAM decoding and latching functions for cache memory subsystems. With its transparent output latches, the 85C508 can decode the tag RAM address, then latch the decoded selects faster than alternative approaches. Figure 7 shows an implementation of the 85C508 as a tag RAM decoder from a high-speed system design. (See the "Acknowledgements" Section for the source of this design.)

In this example, an 85C508 is performing decoding and latching for eight high-speed tag RAM devices. During each memory read, the tag section of the processor's memory address is latched by the address latches to be made available to the tag RAM devices. The address is also decoded by the logic array in the 85C508 and latched by its output latches to provide tag RAM selects. The MISS# signal from the selected tag RAM is driven high if the information at the requested address is in cache memory. MISS# signals are low when the information is not in cache memory (to indicate that a DRAM cycle is required). All MISS# signals are ORed together to generate the active high MISS signal to the processor, which can then proceed with a DRAM cycle. Tag RAM data is written and read on data lines D0–D7.

Figure 8 shows the timing associated with this application based on a 33-MHz clock frequency. A requirement of this design is that a cache hit/miss be resolved in a single clock cycle. This means that the entire decode/access/compare must be performed in 35 ns! (The processor in this application makes the address available 5 ns before the start of the cycle. This pipelining allows a 35 ns access time to be used with the 30 ns clock period.) This is pushing technology to its limits, but the 85C508 is up to the challenge.

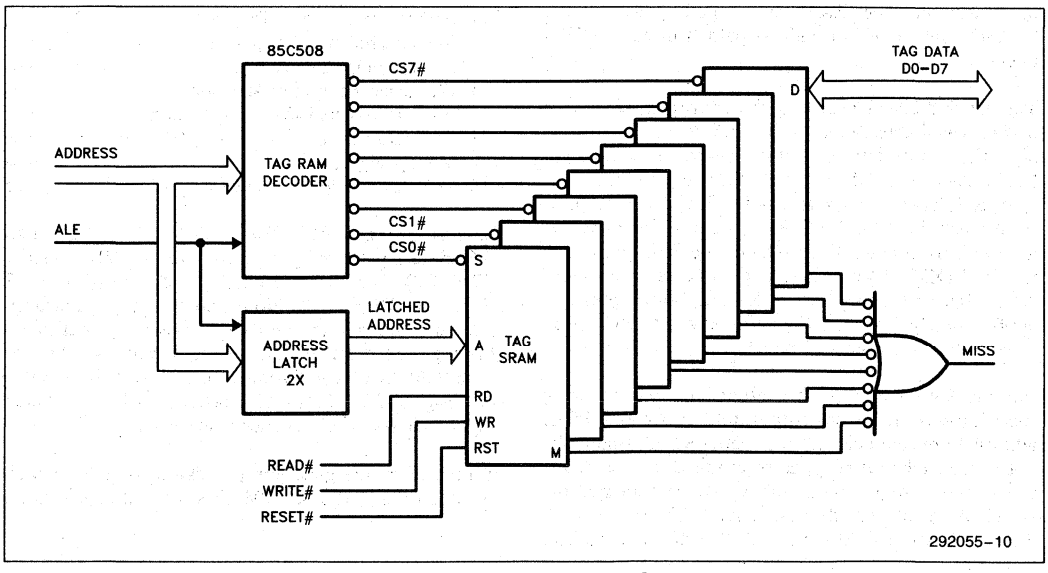


Figure 7. 85C508 Used in a Cache Subsystem

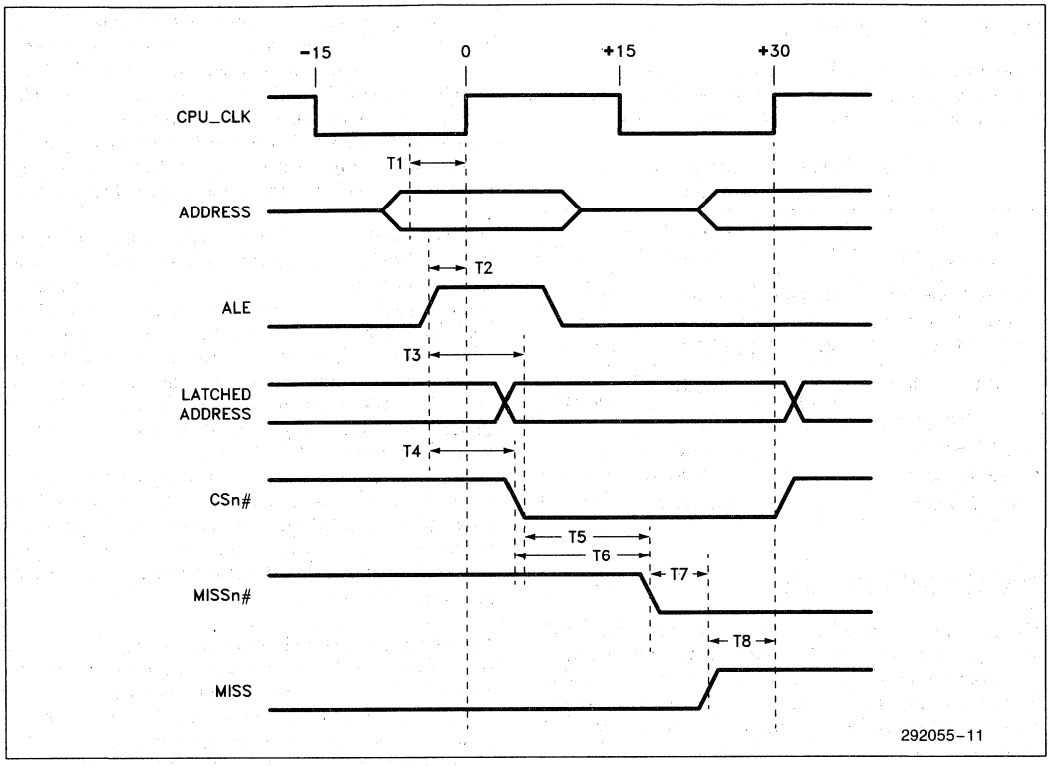


Figure 8. 85C508 Timing in Cache Subsystem

The address for each cycle is made available 5 ns before the rising edge of CPU_CLK; ALE is generated 3 ns before CPU_CLK. Two paths to the tag RAMs must be considered: the RAM select path (through the 85C508) and the RAM address path (through 74X373 latches).

The 85C508 begins decoding the address as soon as it becomes available. No later than 5.5 ns after ALE goes high, the appropriate chip select goes low to select a tag RAM device (the 7.5 ns decode time for the 85C508 is split as follows: 2 ns before ALE high and 5.5 after ALE high). The RAM address is valid to the tag RAM devices 6 ns after ALE goes high (the 8 ns delay time for the 74X373 latches is split as follows: 2 ns before ALE high and 6 ns after ALE high). ALE goes low approximately 6 ns after the rising edge of CPU_CLK to latch the addresses and selects.

The tag RAMs used in the design have the following specifications: 13 ns select-to-output delay and 15 ns address-to-output delay. If the requested information is not in the cache memory, as determined by a comparison between the tag data and tag RAM contents, the MISSn# signal from the respective RAM goes low. A 5 ns delay through the OR gate (MISS gate) and an additional 5 ns processor setup time for the MISS signal completes the timing for the cycle. The individual delays through the circuit are as follows:

T1	Address valid before CPU_CLK high	=	5 ns
T2	ALE high before CPU_CLK high	=	3 ns
T3	Address delay after ALE high	=	6 ns
T4	Chip select delay after ALE high	=	5.5 ns
T5	Address to output valid delay (MISSn#)	=	15 ns
T6	Chip select to output valid delay (MISSn#)	=	13 ns
T7	MISS gate delay	=	5 ns
T8	Processor setup time (MISS signal)	=	5 ns

Measured from CPU_CLK high to CPU_CLK high, the address path delay is 28 ns ((T3 − T2)) + T5 + T7 + T8). The select path delay is 27.5 ns ((T4 − T2) + T6 + T7 + T8). Note that it is the transparent latches coupled with the decode capability that makes the high-speed select path in this circuit possible. No other CMOS PLD solution would allow this circuit to be designed for a 33-MHz processor without introducing a wait state for the cache subsystem.

With caches designed for processors that have more relaxed timing, the 85C508 allows the designer to use

slower SRAM devices to reduce the cost of the cache memory subsystem without sacrificing performance (relative to the processor).

CASCADING DEVICES

More than one 85C508 can be cascaded to provide general-purpose combinatorial logic that can solve speed problems in critical areas of a system. Figure 9 shows one possible configuration. Two outputs from one 85C508 feed a second 85C508. This increases the potential number of inputs to the second device to a total of 30 (16 to the first device and 14 to the second). The ALE signals to both devices are tied high to hold the outputs in transparent mode. The combined propagation delay through both devices is 15 ns max. if 85C508-7s are used.

If some of the outputs from the one device do not need to be routed through both devices, these outputs can be used to provide faster control signals to other areas of the system (7.5 ns max. propagation delay through a single device).

In some cases, cascading devices can also solve latched decoding problems. For example, the configuration shown in Figure 10 can be used as part of a decoder circuit with both ALE signals tied together and driven by bus control circuitry. Here the outputs from the first device select resources that require a faster bus cycle, such as cache RAM. The outputs from the second device select resources that do not require or would not benefit from the faster selects. Extremely fast I/O ports or resources that require wait states anyway, could use a slower select.

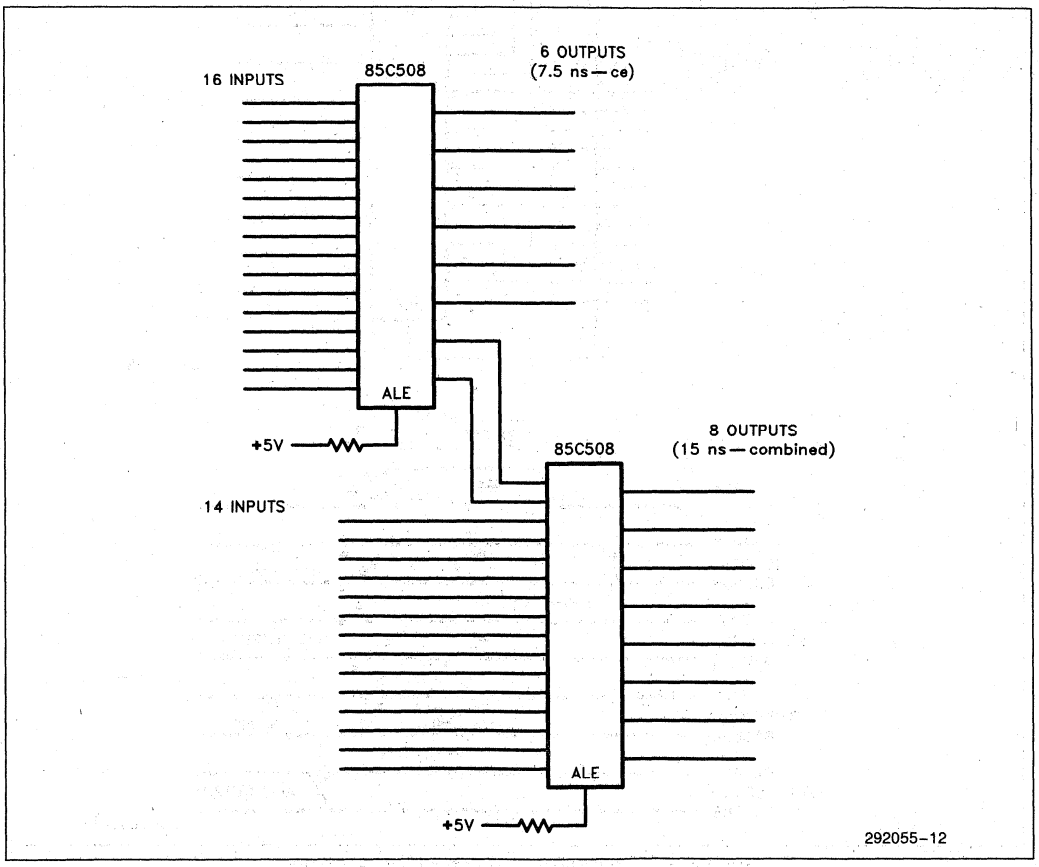
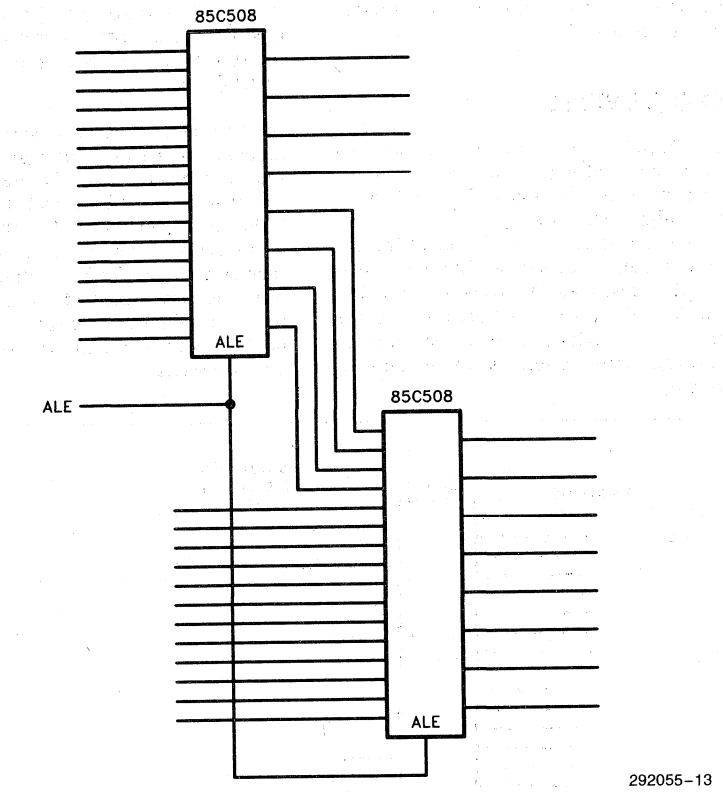


Figure 9. Cascaded 85C508s for Combinatorial Logic



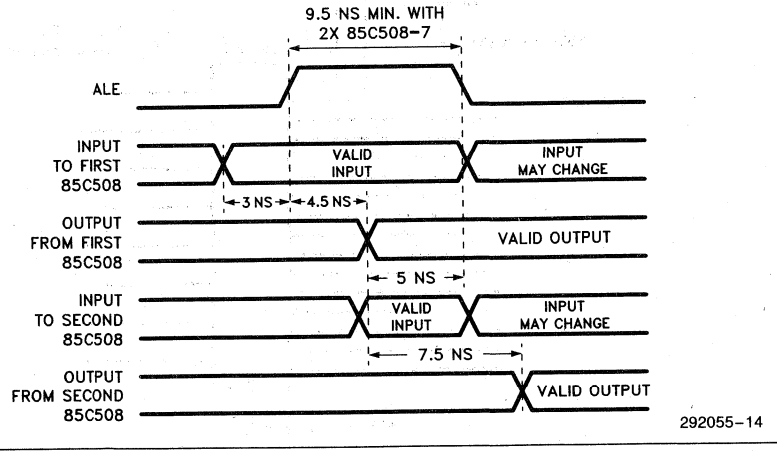


Figure 10. Cascaded 85C508s for Latched Signals

Inputs to the first device must meet the implied setup-to-ALE-high time (3 ns) in order to achieve the specified ALE-to-output time (t_{EO} = 4.5 ns). ALE must be held high for 5 ns after the outputs from the first device become active to meet the setup-to-ALE-low time (t_{SU}) for the second device. These two parameters (t_{EO} and t_{SU}) define the minimum ALE high time for two cascaded 85C508-7 devices at 9.5 ns. Inputs to both devices must also be held until 2 ns before ALE goes low to meet the inpuhold-to-ALE-low time (t_H).

The additional system inputs to the second device (those that do not come from the first device) must meet the same 5 ns t_{SU} and −2ns t_H times as the inputs from the first 85C508. 7.5 ns after the inputs to the second device are valid, the outputs from the second device become valid. Thus there is a fast path for signals that travel through the first or second device *only* and a slightly slower path for signals that travel through *both* devices.

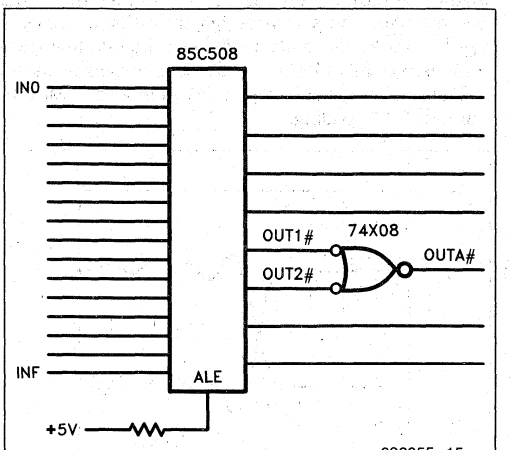
Multiple P-Term Equations

With a single NAND p-term available for logic functions, it is not possible to implement multiple p-term equations such as an OR function with a single output. Multiple p-term functions can be implemented, however, by decoding each p-term separately in parallel and routing the appropriate outputs through an external AND gate (functioning as a DeMorgan's equivalent active-low input NOR gate). The tradeoff for this approach is the additional delay required by the external gate and the use of an additional output in the 85C508. This solution is not practical when several equations require multiple p-terms, but it can provide an ideal solution when only one or two equations require multiple p-terms. Figure 11 shows a configuration where one signal requires a two p-term equation. The maximum delay in this case is 13.8 ns (7.5 ns through the 85C508 and 6.3 ns through the 74X08).

If two inputs to the 85C508 are available and an active high output is needed, OUT1# and OUT2# can be fed back through the device and routed to another output pin. (A NAND/NAND cannot generate an active low output by OR'ing two equations.) In this case, the maximum delay is 15 ns.

NOTE

When using Intel's LOC (Logic Optimizing Compiler) to generate the JEDEC file, the LOC minimizer is often able to reduce equations that seem to require multiple p-terms to a single p-term equation. If an equation cannot be reduced to a single p-term for 85C508 designs, the LOC will abort with an error message. By changing the PART: field in the source ADF to a device with that supports multiple p-terms, and recompiling the ADF with the -L option (generate an LEF—Logic Equation File), you can obtain an output file that shows the lowest possible p-term count. You can then edit the LEF to spread the p-terms across more than one 85C508 output, change the PART: field to 85C508, rename the LEF, and submit it to the LOC again.



ORIGINAL EQUATION:

```
OUTAeqn' = IN8 * IN9 * INA * INB * /INC * /IND *
           /INE * /INF + /IN8 * IN9 * INA *
           /INB * /INC * /IND * INE * INF;
```

PARTITIONED EQUATION:

```
OUT1eqn' = IN8 * IN9 * INA * INB * /INC * /IND *
           /INE * /INF;
OUT2eqn' = /IN8 * IN9 * INA * /INB * /INC * /IND *
           INE * INF;
```

Figure 11. Two P-term Equation Implemented with External AND Gate

BOARD DESIGN INFORMATION

This section discusses information on the 85C508 μPLD that will help to produce a reliable high-speed design.

Output Buffers

The output buffers on the 85C508 are designed to provide fast rise and fall times with reduced signal overshoot and undershoot (as compared to competing CMOS products). Transitions are in the 1–2 ns range (low-to-high and high-to-low). Noise is reduced by using symmetrical N-channel transistors for pullup and pulldown. By preventing a transistor from turning on until the opposing transistor is almost off, noise caused by crossover current is greatly reduced. An additional reduction on overshoot for low-to-high transitions is

accomplished by using a P-channel transistor in parallel with the N-channel pullup transistor. The stronger N-channel transistor pulls the output hard to a TTL-level (2.5V), at which point it shuts off. The weaker P-channel transistor continues to pull the output up to a CMOS level (near 5V) with a minimum of overshoot. Figure 12 shows a typical 85C508 output transition. The overshoot on a low-to-high transition is typically V_{CC} + 0.4V; the undershoot on a high-to-low transition is typically GND − 0.6V. The waveform shown in Figure 12 was made with a 341 ohm load resistor to simulate TTL loading.

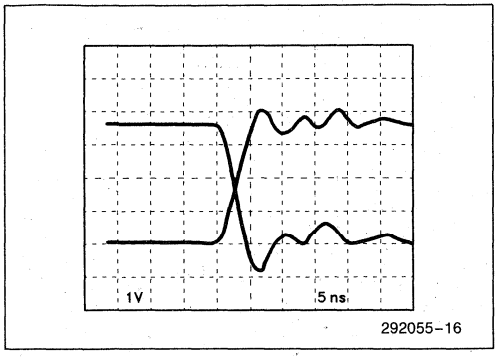


Figure 12. Waveform of Typical 85C508 Output Transition—TTL Levels

Layout Considerations

The V_{PP} (programming) pin on the 85C508 must be tied to ground during normal operation. When tied high or left floating, device operation will be indeterminate.

To minimize I_{CC}, all unused input pins should be tied high or low, but they should not be left floating. Unused output pins can be left floating (unused outputs are held low by the output buffers).

A high-frequency 0.1mF decoupling capacitor is recommended for most applications (X7R or NPO type for stability throughout the operating temperature range). Connect the capacitor between V_{CC} and GND. If the number of outputs switching simultaneously in your application exceeds four, the value of the capacitor should be increased to 0.2 μF.

When using the 85C508 for high-speed decode, it is best to keep the device located as close as possible to the devices it will drive. Short traces reduce both the chance of crosstalk on the outputs lines and the effect of trace impedance on signal transitions.

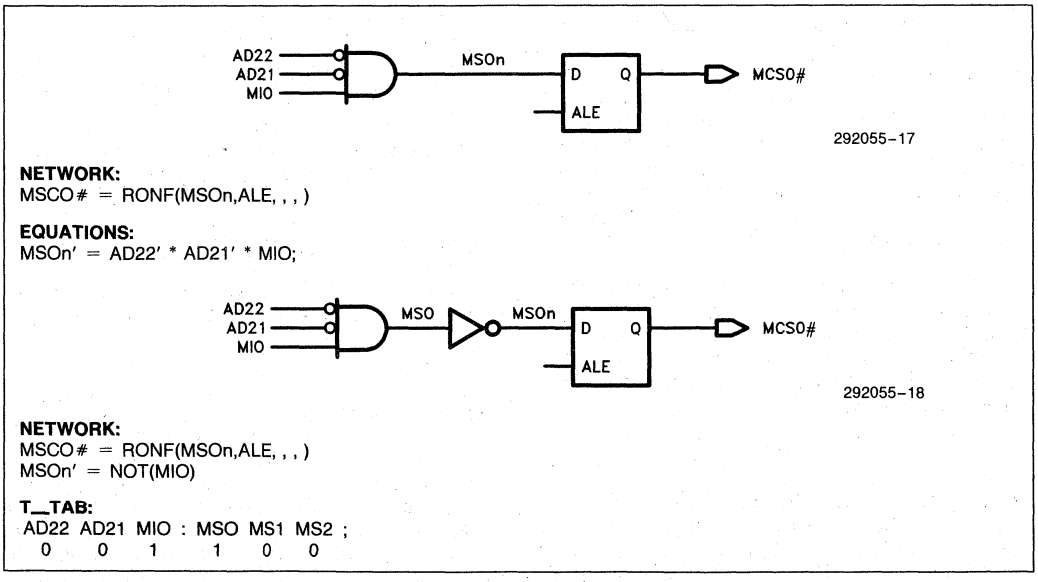


Figure 13. Comparison of ADF and SMF Implementation of NAND P- Term

DESIGN ENTRY

There are two entry methods for implementing an 85C508 design using the iPLS II LOC (Logic Optimizing Compiler): ADF (Advanced Design File) format, or SMF (State Machine File) format. Both methods begin with a text file created by an ASCII text editor.

With ADF format, EPLD design primitives and Boolean equations define the design. An I/O primitive configures each output while an equation is used to represent the NAND p-term (see Figure 13). Figure 14 shows an ADF for a sample decoder. Note that the ADF primitive used to specify the latched outputs on the 85C508 is an RONF (Registered Output—No Feedback). The LOC treats 85C508 output latches as registers, but actual device implementation is a transparent latch. The major task with ADF format is to get the equations into the proper NAND form.

With SMF format, a truth table along with EPLD design primitives define the design. A simple truth table defines the input conditions that drive the output active (high). An I/O primitive configures each output (as with ADFs). The iSTATE converter generates active-high equations as it translates the SMF into an ADF. All combinations of input conditions not explicitly listed in the truth table are automatically resolved to a

```
DANIEL E. SMITH
INTEL CORPORATION
2/3/89
1
A
85C508
508SAMP1: ADF FOR SAMPLE 85C508 DECODER

PART: 85C508

INPUTS:     ALE,                              % global latch enable %
      A2, A3, A4, A5, A6, A7, A8,             % low addr %
      A16, A17, A18, A19, A20, A21, A22       % high addr %
      WR/RD#, M/IO#                           % commands %

OUTPUTS:
      MCS0#, MCS1#, MCS2#,                    % memory selects 0-2 %
      ICS0#, ICS1#, ICS2#, ICS3#, ICS4#       % I/O selects 0-4 %

NETWORK:
      ALE = INP(ALE)
      A2 = INP(A2)
      A3 = INP(A3)
      A4 = INP(A4)
      A5 = INP(A5)
      A6 = INP(A6)
      A7 = INP(A7)
      A8 = INP(A8)
      A16 = INP(A16)
      A17 = INP(A17)
      A18 = INP(A18)
      A19 = INP(A19)
      A20 = INP(A20)
      A21 = INP(A21)
      A22 = INP(A22)
      WRD = INP(WR/RD#)
      MIO = INP(M/IO#)

      MCS0# = RONF(MS0n,ALE,,,)
      MCS1# = RONF(MS1n,ALE,,,)
      MCS2# = RONF(MS2n,ALE,,,)
      ICS0# = RONF(IS0n,ALE,,,)
      ICS1# = RONF(IS1n,ALE,,,)
      ICS2# = RONF(IS2n,ALE,,,)
      ICS3# = RONF(IS3n,ALE,,,)
      ICS4# = RONF(IS4n,ALE,,,)
```

292055-19

Figure 14. Sample ADF for an 85C508 Design

logic low on the outputs. Because of this assumption concerning don't care conditions, truth tables in SMFs should be only used to specify the AND portion of the NAND p-term. The inversion is implemented by routing each truth table output node through a NOT primitive in the Network section of the SMF. This is shown in the example in Figure 13 and in the sample SMF in Figure 15.

```
EQUATIONS:

      MS0n' = A22' * A21' * MIO;
      MS1n' = A22' * A21 * MIO;
      MS2n' = A22 * A21' * MIO;

      IS0n' = A22' * A21' * A20' * A19' * A18' * A17' * A16' * A8'
            * A7' * A6' * A5' * A4 * A3' * A2' * MIO';

      IS1n' = A22' * A21' * A20' * A19' * A18' * A17' * A16' * A8'
            * A7' * A6' * A5 * A4 * A3' * A2 * WRD * MIO';

      IS2n' = A22' * A21' * A20' * A19' * A18' * A17' * A16' * A8'
            * A7' * A6 * A5' * A4 * A3 * A2' * MIO';

      IS3n' = A22' * A21' * A20' * A19' * A18' * A17' * A16' * A8
            * A7' * A6' * A5' * A4 * A3 * MIO';

      IS4n' = A22' * A21' * A20' * A19' * A18' * A17' * A16' * A8
            * A7 * A6 * A5' * MIO';

END$
```

292055-20

Figure 14. Sample ADF for an 85C508 Design (Continued)

```
DANIEL E. SMITH
INTEL CORPORATION
2/3/89
1
A
85C508
508SAMP1: SMF FOR SAMPLE 85C508 DECODER

PART:  85C508

INPUTS: ALE,                                          % global latch enable %
        A2, A3, A4, A5, A6, A7, A8,                   % low addr %
        A16, A17, A18, A19, A20, A21, A22,            % hi addr %
        WR/RD#, M/IO#                                 % commands %

OUTPUTS:
        MCS0#  MCS1#, MCS2#,                          % memory selects 0-2 %
        ICS0#, ICS1#, ICS2#, ICS3#, ICS4#             % I/O selects 0-4 %

NETWORK:
        WRD = INP(WR/RD#)
        MIO = INP(M/IO#)

        MCS0# = RONF(MS0n,ALE,,,)
        MCS1# = RONF(MS1n,ALE,,,)
        MCS2# = RONF(MS2n,ALE,,,)
        ICS0# = RONF(IS0n,ALE,,,)
        ICS1# = RONF(IS1n,ALE,,,)
        ICS2# = RONF(IS2n,ALE,,,)
        ICS3# = RONF(IS3n,ALE,,,)
        ICS4# = RONF(IS4n,ALE,,,)

        MS0n = NOT(MS0)
        MS1n = NOT(MS1)
        MS2n = NOT(MS2)
        IS0n = NOT(IS0)
        IS1n = NOT(IS1)
        IS2n = NOT(IS2)
        IS3n = NOT(IS3)
        IS4n = NOT(IS4)

T_TAB:  % memory selects %

    A22 A21 MIO :  MS0 MS1 MS2 ;
    0   0   1   :  1   0   0   ;
    0   1   1   :  0   1   0   ;
    1   0   1   :  0   0   1   ;

T_TAB:  % I/O selects 0-4 %

   A22 A21 A20 A19 A18 A17 A16 A8 A7 A6 A5 A4 A3 A2 WRD MIO : IS0 IS1 IS2 IS3 IS4 ;
    0   0   0   0   0   0   0   0  0  0  0  1  0  0  X   0  :  1   0   0   0   0  ;
    0   0   0   0   0   0   0   0  0  0  1  1  0  1  1   0  :  0   1   0   0   0  ;
    0   0   0   0   0   0   0   0  0  1  0  1  1  0  X   0  :  0   0   1   0   0  ;
    0   0   0   0   0   0   0   1  0  0  0  1  1  X  X   0  :  0   0   0   1   0  ;
    0   0   0   0   0   0   0   1  1  1  0  X  X  X  X   0  :  0   0   0   0   1  ;

END$
```

292055-21

Figure 15. Sample SMF for an 85C508 Design

SUMMARY

The 85C508 μPLD provides a high-performance, low-power solution for high-speed microcomputer systems. The 85C508 allows designers to achieve the maximum performance from today's generation of high-speed processors and provides enough speed to support the needs of tomorrow's generation of processors.

ACKNOWLEDGEMENTS

A special thanks to Larry Jubb of Wideband Logic Design (San Jose, CA) and Daisy Systems Corp. for making available the SPARC-based cache circuit Larry designed for Daisy's Gigalogician project.

September 1990

3

In-Circuit Reconfiguration of the 85C960 and 85C508 μPLDS

THOM BOWNS
PROGRAMMABLE LOGIC APPLICATIONS

Order Number: 292072-001

IN-CIRCUIT RECONFIGURATION OF THE 85C960 AND 85C508 μPLDS

CONTENTS

INTRODUCTION

The 85C960 and 85C508 are application-specific μPLDs (Microcomputer Programmable Logic Devices) designed to work synergistically with today's high-speed microprocessors and microcontrollers. These μPLDs are based upon a state-of-the-art, patented Intel CHMOS* technology, making them very fast while maintaining flexibility and low power and heat dissipation.

These devices use CMOS EPROM technology for non-volatile storage of their configuration. The devices are configured initially by compiling a Boolean equation source file using iPLS II, and programming the EPROM cells. During device power-up, this configuration is loaded into a volatile architecture array. It is this volatile architecture array that enables in-circuit reconfiguration.

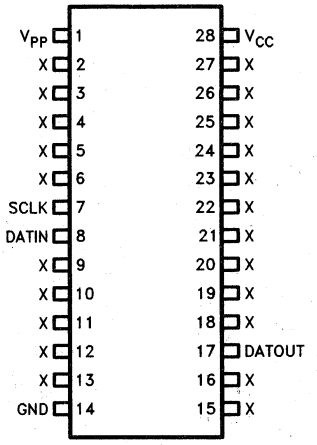
By placing one of these devices into reconfiguration mode, this volatile architecture configuration array can be read and/or modified, making it possible to dynamically reconfigure the portions of the devices, without actually programming or erasing the EPROM cells. An external circuit that includes FIFO memory is required to store new configuration data, place the μPLDs into the reconfigure mode, and shift in the new configuration.

RECONFIGURABILITY

Both the 85C0960 and 85C508 use the same basic method for reconfiguration; only the bit maps differ. Figure 1 shows the pins used and voltages required to reconfigure the devices. Figure 2 shows the reconfiguration waveforms and timing.

The shift order for the 85C508 is a simple, sequential, highest-bit-in-first data shift. Thus, labeling the JEDEC bits from 0 to 255, bit 255 goes in first, followed by bit 254, then bit 253, and so on to bit 0. The JEDEC shift order for the 85C960, however, is not sequential. Table 1 shows the shift order for the 85C960.

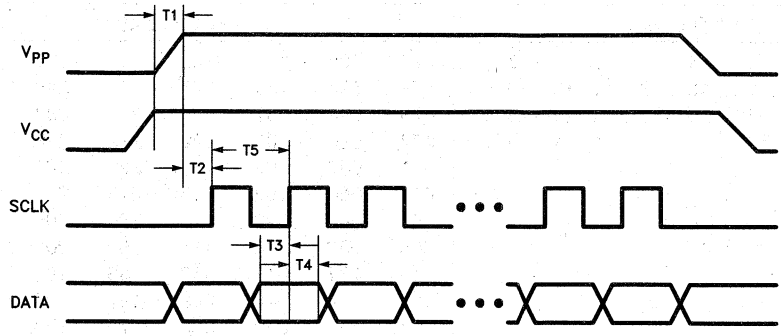
To place these devices into reconfiguration mode, V_{CC} is applied to the device and pin 1 is then raised to 12.5V. Data is introduced serially on pin 8, and is shifted in by clocking pin 7. When pin 1 is returned to a normal CMOS level, the device resumes operation using the new configuration. Note that pin 1 on the 85C960 is a RESET pin during normal operation. With V_{CC} at a nominal level (4.75V–5.25V), a system reset (low pulse on pin 1) does not affect the volatile configuration. For both devices, however, when V_{CC} drops below 1.2V, an internal reset occurs and the nonvolatile EPROM cells are re-loaded into the volatile configuration array; i.e., the device returns to its originally programmed configuration.



292072-1

Parameter	Description	Units
V_{PP}	Special Mode Supervoltage	12.5V
V_{CC}	Supply Voltage	5.0V
DATIN	Serial Input Data	0V or 5V
DATOUT	Serial Output Data	0V or 5V
SCLK	Serial Shift Clock	0V or 5V
X	Don't Care	0V or 5V

Figure 1. 85C960/85C508 Special Mode Pins/Voltages



Parameter	Description	Value	Units
T1	V_{CC} to V_{PP} Min	250	ns
T2	V_{PP} to CLK Min	500	ns
T3	Data Setup to CLK Min	100	ns
T4	CLK to Data Hold Min	100	ns
T5	CLK Period Min	250	ns

Figure 2. Reconfiguration Timing Specifications

Table 1. Shift Order to JEDEC Fuse Mapping for the 85C960 μPLD

Shift Order	Fuse Number							
(1 2 3 4 5 6 7 8)	079	095	111	127	252	253	254	255
(9 10 11 12 13 14 15 16)	078	094	110	126	248	249	250	251
(17 18 19 20 21 22 23 24)	075	091	107	123	236	237	238	239
(25 26 27 • • • • •)	074	090	106	122	232	233	234	235
	071	087	103	119	220	221	222	223
	070	086	102	118	216	217	218	219
	067	083	099	115	204	205	206	207
	066	082	048	114	200	201	202	203
	015	031	047	063	188	189	190	191
	014	030	046	062	184	185	186	187
	011	027	043	059	172	173	174	175
	010	026	042	058	168	169	170	171
	007	023	039	055	156	157	158	159
	006	022	038	054	152	153	154	155
	003	019	035	051	140	141	142	143
	002	018	034	050	136	137	138	139
	001	017	033	049	132	133	134	135
	000	016	032	048	128	129	130	131
	005	021	037	053	148	149	150	151
	004	020	036	052	144	145	146	147
	009	025	041	057	164	165	166	167
	008	024	040	056	160	161	162	163
	013	029	045	061	180	181	182	183
	012	028	044	060	176	177	178	179
	065	081	097	113	296	297	298	299
	064	080	096	112	292	293	294	295
	069	085	101	117	212	213	214	215
	068	084	100	116	208	209	210	211
	073	089	105	121	228	229	230	231
	072	088	104	120	224	225	226	227
	077	093	109	125	244	245	246	247
(• • 253 254 255 256)	076	092	108	124	240	241	242	243

Figure 3 shows an example of the support hardware required for reconfiguration. The circuit consists of a controller that occupies an I/O port and decodes the commands for performing reconfiguration, a FIFO RAM to store the configuration data, and discrete components for high-voltage and three-state switching. The controller is implemented in a 5AC312 EPLD, a high-integration general-purpose device. The source listing for the controller is shown in Figure 4. A microcontroller flowchart for reconfiguring the 85C960 or 85C508 using this circuit provided in Figure 5. (For information on the 5AC312 and the iPLS II software, refer to the Programmable Logic handbook, order number: 296083. The handbook also provides information on programming support, by Intel and major third-party vendors.)

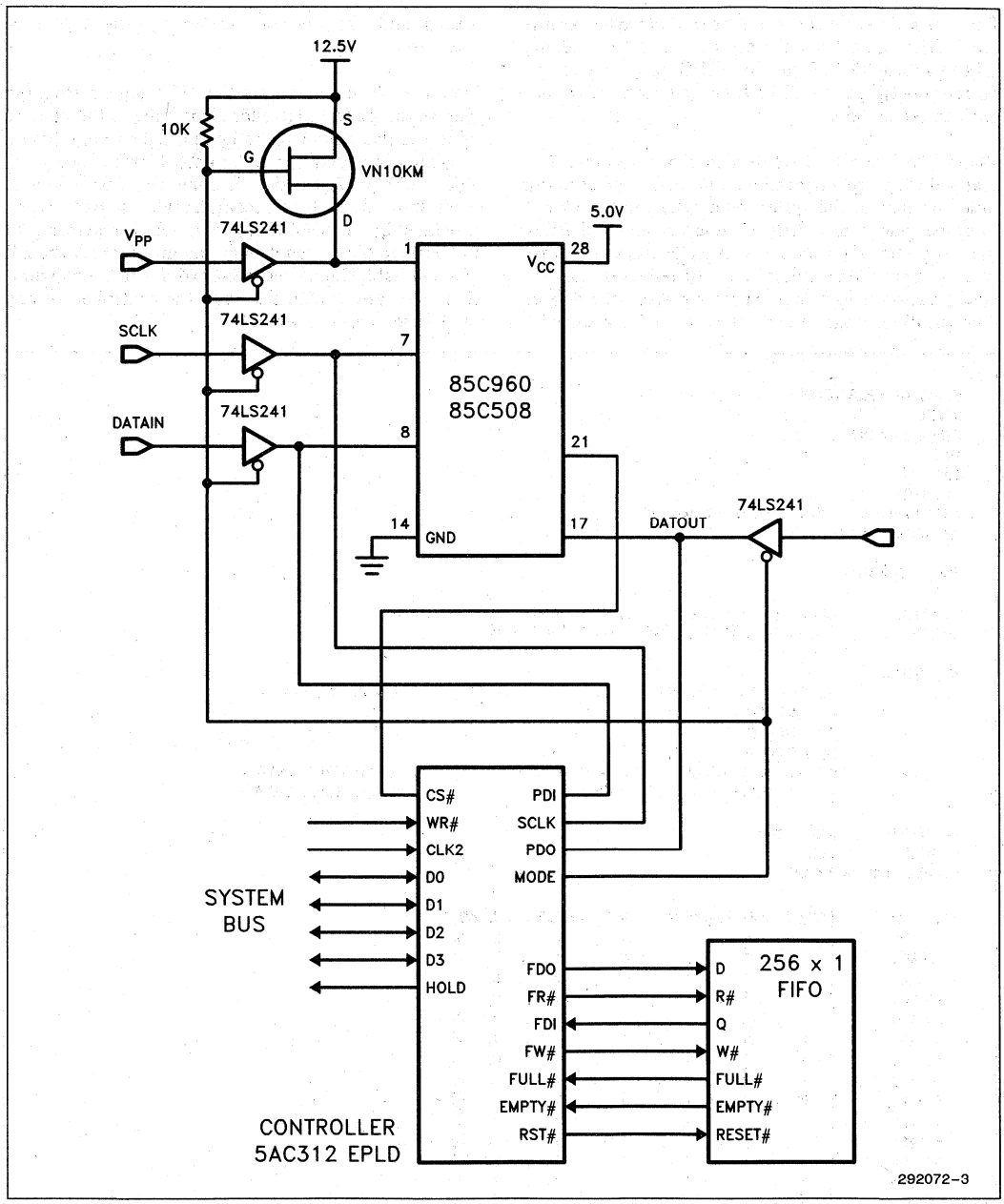


Figure 3. Reconfiguration Circuit

Four operations are supported by the controller: Transfer JEDEC data from the system to FIFO, load the μPLD with JEDEC data from FIFO, read the μPLD's current configuration into FIFO, and read FIFO data back to the system.

The 85C960 and 85C508 devices cannot operate normally during reconfiguration. The processor and any other portion of the system that relies on the μPLD must be placed in a hold or internal loop condition until the reconfiguration is complete. Also, note that when power to the devices is lost or removed, the new configuration is lost. When power returns, the default configuration programmed into the EPROM array is loaded and the devices operate as originally programmed.

Note that the circuit shows the 85C508 generating the chip select for the controller. This reduces the number of selects that can be reconfigured, unless the software is aware of the new controller port. If this is not desirable, the chip select to the controller could be generated from some fixed (not reconfigurable) source. When generated by the 85C508, the chip select, together with the data, initiates a controller command. With the LE (Latch Enable) input to the 85C508 during reconfiguration, no spurious chip selects to the controller, or any other resource, is generated.

```
PLFG APPLICATIONS
INTEL
APRIL 11, 1990
U3
005
5AC312
85C960/508 Reconfiguration circuit controller
OPTIONS: TURBO = ON

PART: 5AC312

INPUTS:     CLK2, nCS, nWR, PDO, FDO, nFULL, nEMPTY, nRD, D1, D2, D3
OUTPUTS:    D0, MODE, PDI, SCLK, FDI, nFRD, nFWR, nFRST

NETWORK:
            D0, D0 = COIF (FDO, RD)                    % BUS I/O DATA %
            D1 = INP(D1)
            D2 = INP(D2)
            D3 = INP(D3)
            PDI = RONF(PDId, CLK2, GND, GND, MODE)     % PLD INPUT DATA %
            FDI = RONF(FDId, CLK2, GND, GND, VCC)      % FIFO INPUT DATA %

MACHINE:    RECONFIG
CLOCK:      CLK2
% OE on sclk is 'MODE' %

STATES:     [SCLK, nFRD, nFWR, nFRST, MODE, MB, XB]
     PU     [  0  ,  0 ,  0  ,   0  ,   0 ,  0,  0]
  RESET     [  1  ,  1 ,  1  ,   0  ,   0 ,  0,  0]
   WAIT     [  1  ,  1 ,  1  ,   1  ,   0 ,  0,  0]
  FTOP1     [  1  ,  1 ,  1  ,   1  ,   1 ,  0,  1]
  FTOP2     [  1  ,  0 ,  1  ,   1  ,   1 ,  0,  0]
  FTOP3     [  0  ,  1 ,  1  ,   1  ,   1 ,  0,  0]
  FTOB1     [  1  ,  1 ,  1  ,   1  ,   0 ,  0,  1]
  FTOB2     [  1  ,  0 ,  1  ,   1  ,   0 ,  0,  0]
  FTOB3     [  0  ,  1 ,  1  ,   1  ,   0 ,  0,  0]
  PTOF1     [  1  ,  1 ,  1  ,   1  ,   1 ,  1,  1]
  PTOF2     [  1  ,  1 ,  0  ,   1  ,   1 ,  1,  0]
  PTOF3     [  0  ,  1 ,  1  ,   1  ,   1 ,  1,  0]
  BTOF1     [  1  ,  1 ,  1  ,   1  ,   0 ,  1,  1]
  BTOF2     [  1  ,  1 ,  0  ,   1  ,   0 ,  1,  0]
  BTOF3     [  0  ,  1 ,  1  ,   1  ,   0 ,  1,  0]
```

292072-4

Figure 4. Reconfiguration Controller Source ADF

```
                    % STATE TRANSITIONS %

PU:         RESET

RESET:      WAIT

WAIT:       IF COMMAND1 THEN FTOP1          % COMMAND 1 IS FIFO TO PLD %
            IF COMMAND2 THEN FTOB1          % COMMAND 2 IS FIFO TO BUS %
            IF COMMAND3 THEN PTOF1          % COMMAND 3 IS PLD TO FIFO %
            IF COMMAND4 THEN BTOF1          % COMMAND 4 IS BUS TO FIFO %

FTOP:       IF !nEMPTY THEN RESET
            FTOP2
FTOP2:      FTOB3
FTOP3       FTOP1

FTOB1:      IF !nEMPTY THEN RESET
            PTOB2
FTOB2:      IF RD THEN FTOB3
FTOB3:      PTOB1

PTOF1:      IF !nFULL THEN RESET
            PTOF2
PTOF2:      PTOF3
PTOF3:      PTOF1

BTOF1:      IF !nFULL THEN RESET
            BTOF2
BTOF2:      IF !nWR THEN BTOF3
BTOF3:      BTOF1


EQUATIONS:
            COMMAND1 = !D3 * !D2 * D1 * !nWR * !nCS;
            COMMAND2 = !D3 * D2 * !D1 * !nWR * !nCS;
            COMMAND3 = !D3 * D2 * D1 * !nWR * !nCS;
            COMMAND4 = D3 * !D2 * !D1 * !nWR * !nCS;

            FDId    = BTOF * D0
                    + PTOF * PDO
                    + FTOB * FDO
                    + FTOP * FDO;

            PDId    = FTOP * FDO
                    + FTOB * PDO
                    + BTOF * PDO
                    + PTOF * PDO;

            RD = !nRD;

            FTOP = MODE * !MB;
            FTOB = !MODE * !MB;
            PTOF = MODE * MB;
            BTOF = !MODE * MB;

END$
```

292072-5

Figure 4. Reconfiguration Controller Source ADF (Continued)

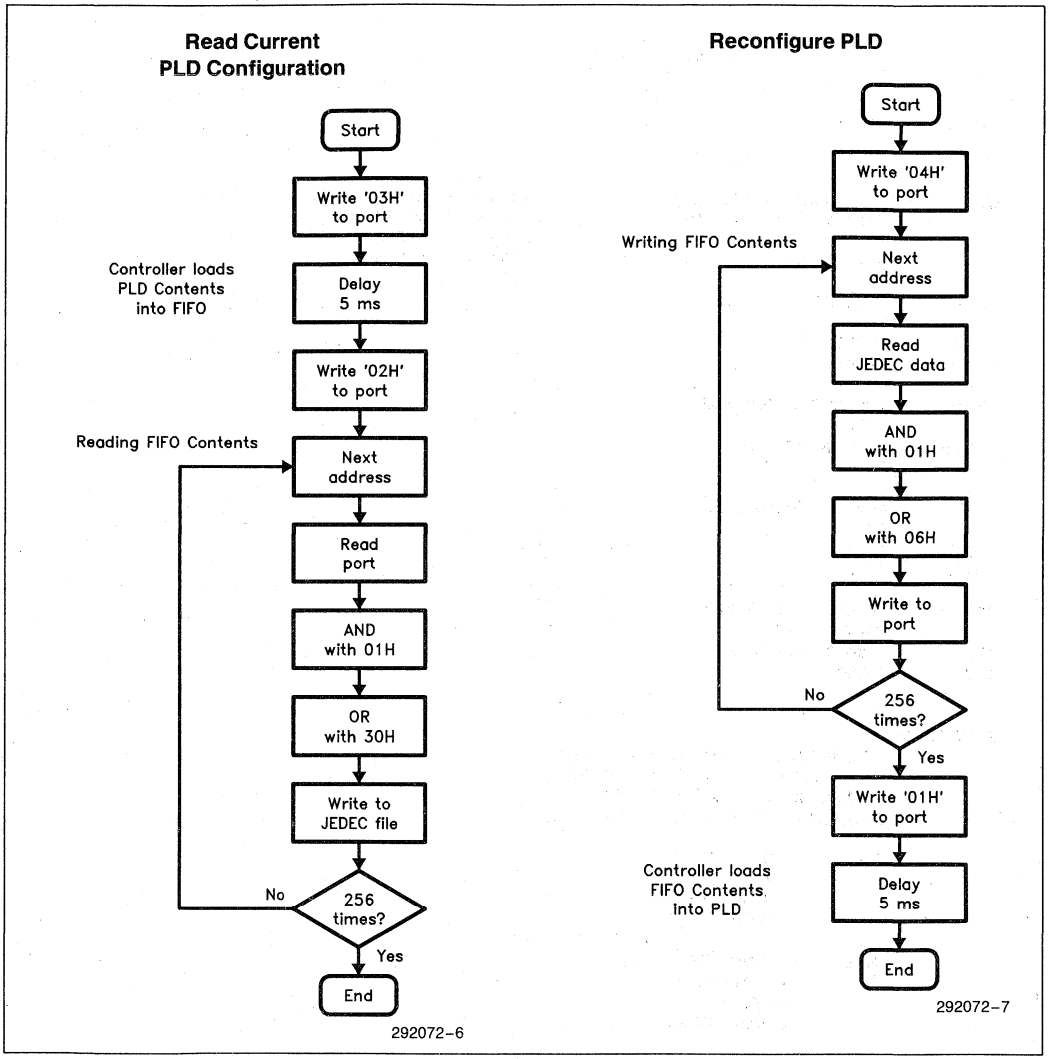


Figure 5. Reconfiguration Controller Protocol Flow Chart

SUMMARY

By reconfiguring an 85C508 or 85C960 μPLD in-circuit, the designer has the flexibility to dynamically change system performance without going through a design iteration or even reprogramming the device. This allows a degree of system configuration—with no down time—not previously possible with standard PLDs.

September 1990

3

85C220/85C224 Design Guide

DANIEL E. SMITH, THOM BOWNS, AND TERRY BAUCOM
INTEL CORPORATION
PROGRAMMABLE LOGIC APPLICATIONS

Order Number: 292073-001

85C220/85C224 Design Guide

CONTENTS

CONTENTS

INTRODUCTION

In the past decade and half, system designers have come to depend on PLDs (Programmable Logic Devices) to implement random logic and interface circuits because of the low cost and high performance these devices offer. Bipolar PLDs (commonly called PALs*) were the first class of devices to become a common part of the designer's repertoire. As power requirements and head dissipation of bipolar PLDs became a design and/or reliability limiter in the mid-1980's, designers began using CMOS PLDs (GALs** and EPLDs).

In 1989 and 1990 Intel began shipping its fast 85C220 and 85C224 PLDs. These PLDs are supersets of common 20-pin and 24-pin PALs/GALs. Both the 85C220 and 85C224 exhibit extremely fast speeds, low power consumption, and high integration, which make them ideally suited for high-speed microcomputer system applications. This application note covers the architectural features that distinguish these Intel PLDs (also called μPLDs for Microcomputer Programmable Logic Devices) from their competition. It also covers performance, power requirements, and heat dissipation. Finally, it describes electrical characteristics of the devices, along with guidelines for taking best advantage of device behavior while mitigating tradeoffs that may be encountered.

PERFORMANCE

PLDs can be measured in terms of propagation delay (t_{PD}) or state machine frequency ($1/t_{SU} + t_{CO}$). In terms of propagation delay, performance of the first CMOS PLDs lagged behind bipolar PLDs by 15 ns–20 ns. With the 85C220/85C224, this gap has shortened to only 2.5 ns. In terms of state machine frequency, CMOS PLDs also historically lagged bipolar PLDs significantly. During late 1989, however, the state-machine performance of Intel PLDs actually surpassed bipolar devices. Table 1 lists the key speed parameters of 16-series and 20-series PALs and GALs, and the 85C220/85C224. The additional performance provided by the 85C220/85C224 can allow faster state machine designs or provide additional margin for existing designs.

Table 1. PAL/GAL/85C220/85C224 Performance Comparison

Parameter	16R8-D 20R8-D	16R8-E 20R8-E	16V8-10 20V8-10	85C220-80 85C224-80	Units
t_{PD}	10	*7.5*	10	10	ns
t_{SU}	10	*7*	10	*7*	ns
t_{CO}	7	6.5	8	*5.5*	ns
f_{CNT1}	58.8	74	55.5	*80*	MHz
t_{CNT}	16.5	10	Not Avail.	*10*	ns
f_{CNT2}	60	*100*	Not Avail.	*100*	MHz
f_{MAX}	62.5	100	62.5	*111*	MHz

Italics indicate fastest specification for that category.

Parameter Definitions:

t_{PD} — Propagation Delay, Input or I/O to Output Valid Delay
t_{SU} — Input or I/O Setup Time to Clock
t_{CO} — Clock to Output Valid Delay
f_{CNT1} — Max. External Counter Frequency ($1/t_{SU} + t_{CO}$)
t_{CNT} — Clock to Feedback Setup for Next Clock—Internal Path
f_{CNT2} — Max. Internal Counter Frequency ($1/t_{CNT}$)
f_{MAX} — Max. Pipelined Frequency

*PAL® is a registered trademark of Advanced Micro Devices.
**GAL® is a registered trademark of Lattice Semiconductor, Inc.

ARCHITECTURAL SUPERSET FEATURES

This section summarizes the architectural limitations of PALs and GALs and describes the superset features of the 85C220/85C224 devices.

Standard PALs and Their Limitations

Designers are familiar with common 16-series and 20-series PAL and GAL devices. The layout is standard with clock, output enable, inputs and outputs in the same pin locations across families of devices (all signals on R4, R6, and R8 devices; inputs and outputs on L8 devices). Seven or eight p-terms (product-terms) are available for logic implementation. The major task with PALs is to pick the device with the desired range of inputs and registers. The designer then fits the design into the device, modifying the design to fit the architecture of the device.

Architectural limitations that experienced PAL users have all encountered include:

1. One active output polarity per device (usually active low)
2. Only 7 SOP (Sum-of-Products) p-terms on combinatorial outputs
3. Unusable pin on R4, R6 and R8 devices if the dedicated OE is not used
4. No feedback on the outside macrocells (#0 and #7) of L8 devices
5. Wasted resources in cases where the number and type of outputs does not match device resources. (For example, a 16R-series or 20R-series PAL cannot implement a design requiring 5-registered and 3-combinatorial outputs. An R6 would be needed to implement the 5 registers and 2 of the 3 combinatorial outputs. One register in the R6 would not be used and the 3rd combinatorial output would have to be implemented in discrete logic or in an additional PAL.)

GALs Provide a Partial Improvement

The GAL architecture overcomes some of these limitations. GALs offer programmable output polarity so that any output can be active high or active low. Any combination of registered/combinatorial outputs can be implemented in a single device. But several architectural limitations are still present in GALs:

1. Only 7 SOP p-terms are available for combinatorial outputs if a p-term is used for an OE signal
2. The dedicated OE pin is unusable if registers are used but the OE control is not needed
3. No feedback on the outside macrocells (#0 and #7) when the device is configured for all combinatorial outputs.

Thus while GALs provide greater flexibility than PALs, GAL users still find themselves designing around the architectural limitations.

85C220/85C224—The Total Solution

With the Intel 85C220/85C224 PLDs, the architectural limitations of PALs and GALs have been overcome. The 85C220 is an architectural superset of 16-series (20-pin) PALs and GALs, while the 85C224 is an architectural superset of 20-series (24-pin) devices. Let's look at the superset features.

SOP Invert

Figure 1 compares the macrocells (output structures) of PALs, GALs, and the 85C220/85C224. Note that the 85C220/85C224 and the GAL contain a programmable invert option to allow each output to be individually configured as active low or active high. This feature allows logic compilers to use DeMorgan's inversion techniques to fit equations into an 85C220/85C224 that would not fit into a standard PAL. Larger equations will fit into the 85C220/85C224 that will not fit in 16-series and 20-series PALs.

Independently Configurable Outputs

The 85C220/85C224 and the GAL allow any combination of registered and combinatorial outputs in a single device. This feature means that designers will not have to leave outputs unused as with PALs.

Output Enable P-Term

Figure 1 also shows that the 85C220/85C224 contains an output enable p-term in addition to the 8 SOP p-terms (9 p-terms total). L8 PALs and GALs borrow one of the SOP p-terms to implement an OE equation, leaving only 7 p-terms to implement the SOP. In cases where an SOP equation requires all 8 p-terms, the PAL/GAL designer must route part of the equation through another macrocell first (and accept the additional delay), or choose a different device. Since the 85C220/85C224 does not have this limitation, larger designs will fit more often in Intel PLDs.

Output Feedback

Note that the two outside macrocells (#0 and #7) on L8 PALs and GALs configured for combinatorial operation do not allow feedback to the logic array. If this feedback is needed, a different device must be chosen.

This may mean switching to an 18P8, 18CV8, etc., or moving up to a larger device with more inputs. All 8 outputs on the 85C220/85C224 can feed back a combinatorial signal from the I/O pin to the logic array. The need to change devices or move up to a larger device is diminished for 85C220/85C224 users.

No Dedicated Output Enable Pin

When using PALs and GALs configured for registered operation, the OE pin can only be used as an OE signal. If no OE function is needed, the pin must be tied high; it cannot be used as a standard input to the logic array. The 85C220/85C224, however, does not contain a dedicated output enable pin, making these devices more flexible than PALs and GALs. With the 85C220/85C224, if the registers do not require an output enable, the internal p-term can hold the output buffers in the enabled state. The corresponding pin can then be used as a dedicated input. If a global OE is needed, then the OE p-terms for all eight macrocells can be programmed identically.

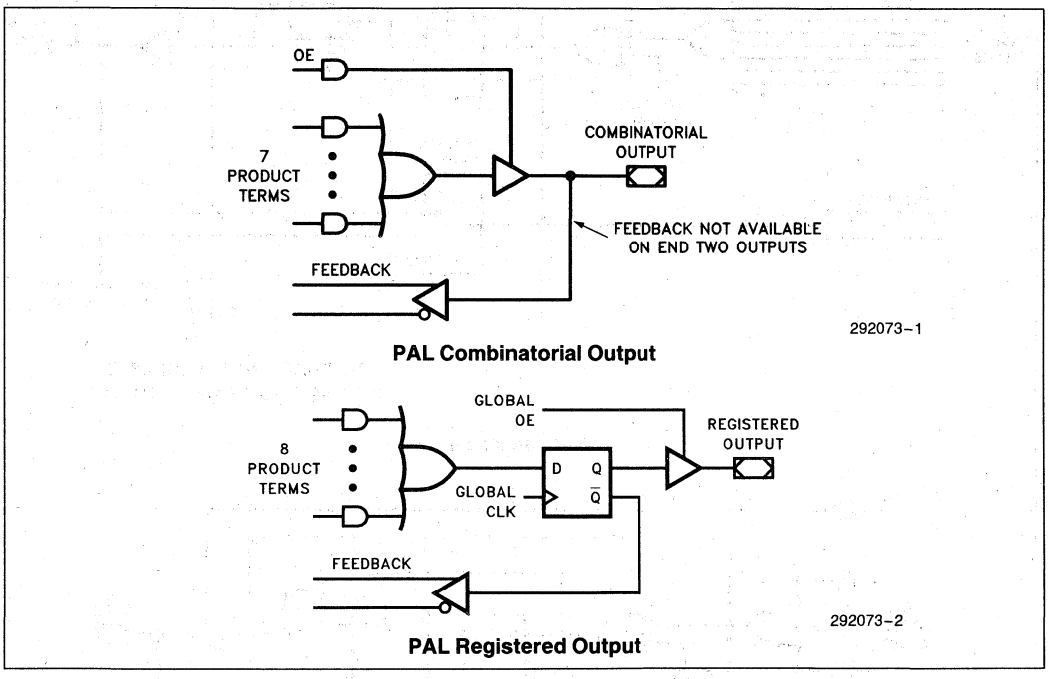
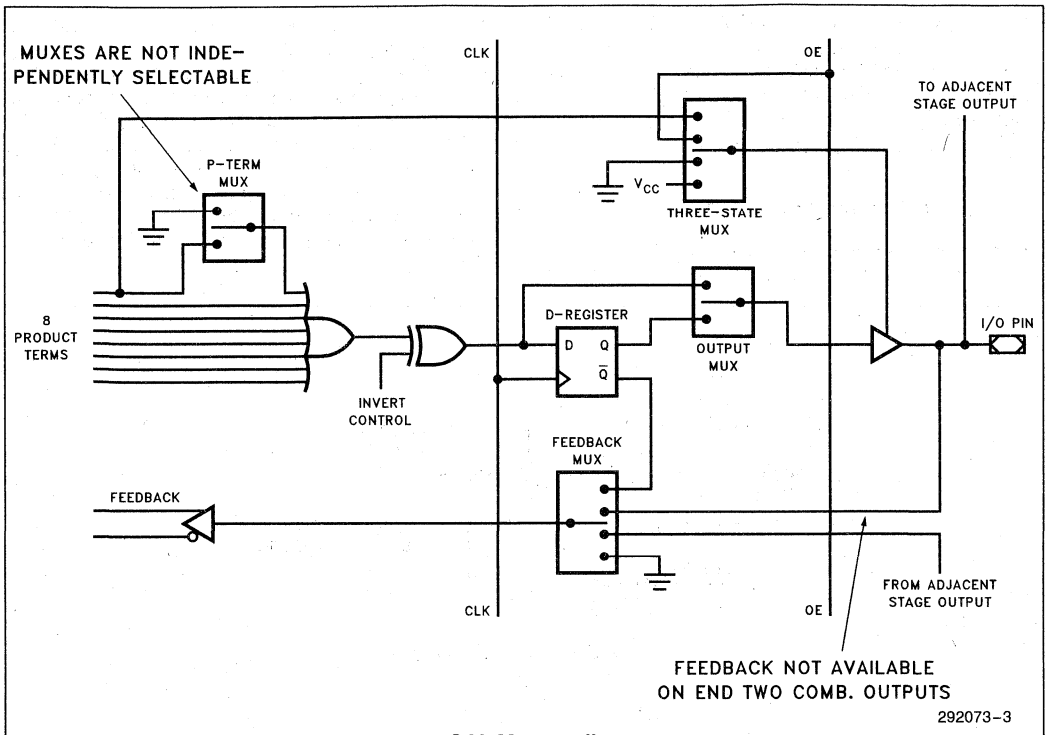
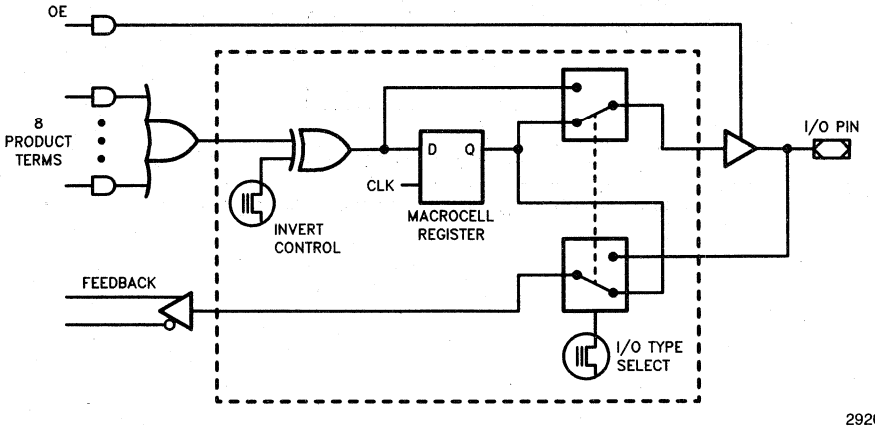


Figure 1. Output/Macrocell Architecture Comparison



GAL Macrocell



85C220/85C224 Macrocell

Figure 1. Output/Macrocell Architecture Comparison (Continued)

Architecture Summary

The combination of PAL/GAL compatibility along with the superset features on the 85C220/85C224 μPLDs allows more logic to be implemented in these devices than in standard PAL/GAL architectures. The need to stock several different PAL/GAL architectures (in multiple packages) is reduced when using the 85C220/85C224 architectures. 85C220/85C224 users will find that their designs outgrow their PLDs significantly less often than PAL/GAL users.

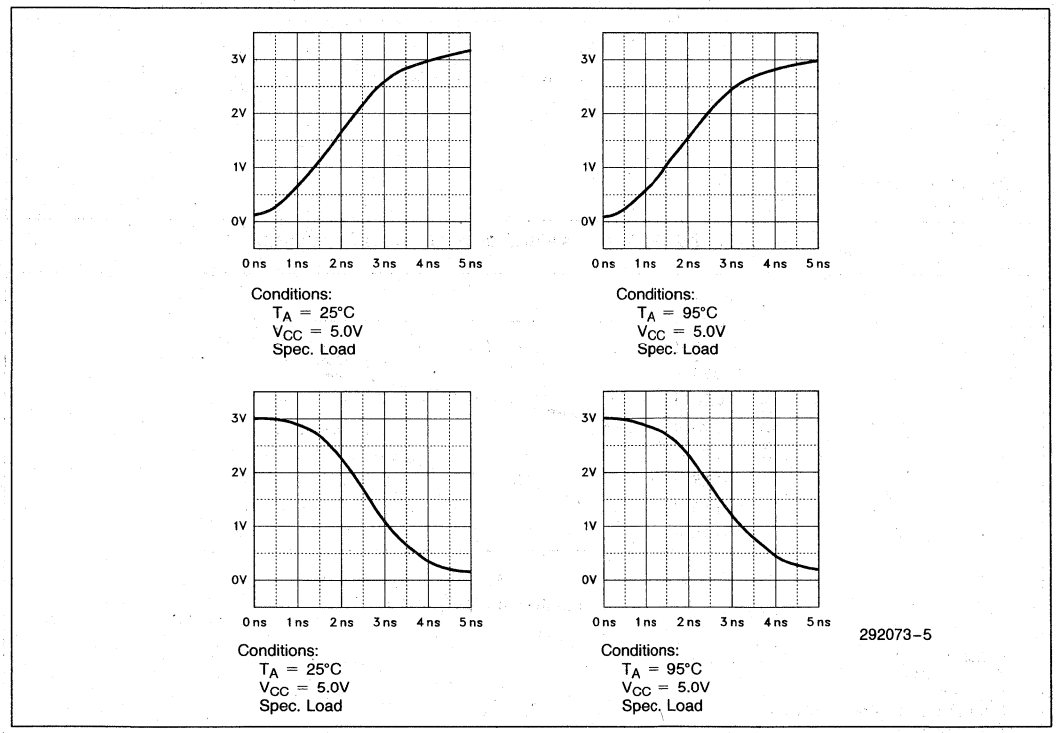
ELECTRICAL CHARACTERISTICS

This section provides characterization data for the 85C220/85C224 that can be of great help to designer's working on high-speed systems. Information is presented in the following order:

- Output Slew Rates
- t_{PD} Characteristics
- t_{CO} Characteristics
- I_{OL} Characteristics
- Tracking Critical Parameters over Temperature
- Ground Bounce/Noise on Unswitched Outputs

Output Slew Rates

Output buffers for 85C220/85C224 devices have a nominal slew rate of 1 V/ns low-to-high and 1.5 V/ns high-to-low. At 95°C, the slew rate slows slightly to 0.8 V/ns low-to-high, and 1.2 V/ns high-to-low. This slew rate is fast enough to meet the requirements of all 85C220/85C224 specifications, but slow enough to minimize problems such as ground bounce and transmission line effects. Figures 2 through 5 show slew rates as measured on 85C220/85C224 devices.



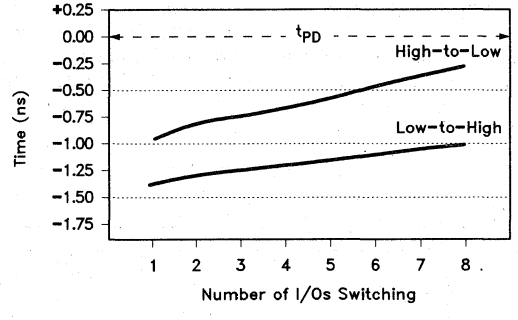
Figures 2–5. 85C220/85C224 Output Slew Rates

t_{PD} Characteristics

This section shows how propagation delay for the 85C220/85C224 PLDs is affected by factors that vary from one application to another.

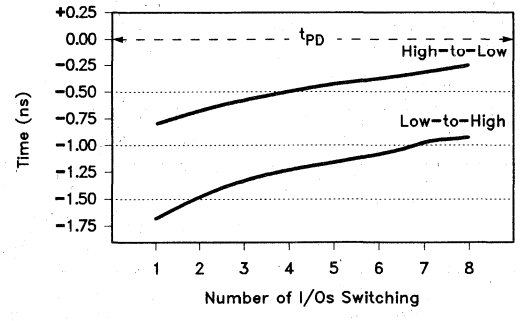
t_{PD} vs Number of Outputs Switching

As the number of device outputs switching simultaneously increases, average propagation delay through devices also increases. This increase is related to the ability of package power and ground leads to channel the additional current. Figures 6 and 7 show the relation of t_{PD} to the number of outputs switching for the 85C220 and 85C224, respectively. Note that this data reflects worst case values. Typical parts can be 1 ns–2 ns faster. This data helps designers estimate how much margin exists in a high-speed design.



Conditions:
T_A = +70°C
V_{CC} = 4.75V
Input and Output Ref. = 1.50V
Spec. Load

Figure 6. 85C220 t_{PD} vs Number of Outputs Switching



Conditions:
T_A = +70°C
V_{CC} = 4.75V
Input and Output Ref. = 1.50V
Spec. Load

Figure 7. 85C224 t_{PD} vs Number of Outputs Switching

t_{PD} vs C_L

Knowledge of how PLDs behave as capacitive loading is increased is an important consideration when designing high-speed systems. Figure 8 shows derating from specified values for a typical 85C220/85C224 at high temperature, low V_{CC} conditions for both low-to-high and high-to-low transitions as capacitance increases. These characteristics can help designers trade off system margin for additional delays incurred due to higher capacitive loads.

t_{PD} vs Number of P-Terms Switching

As additional p-terms for a macrocell are used, internal device loading causes propagation delay to increase or decrease slightly. Figure 9 shows this slowdown for the 85C220/85C224 at room temperature conditions with 2 outputs switching. Data is shown for both low-to-high and high-to-low transitions.

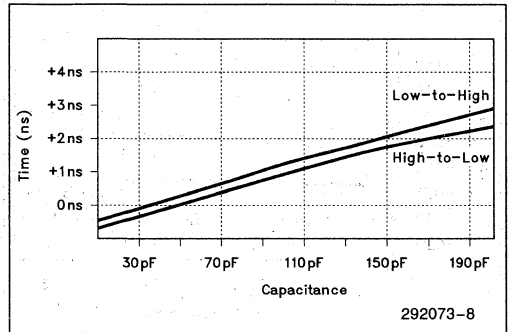


Figure 8. 85C220/85C224 t_{PD} vs C_L

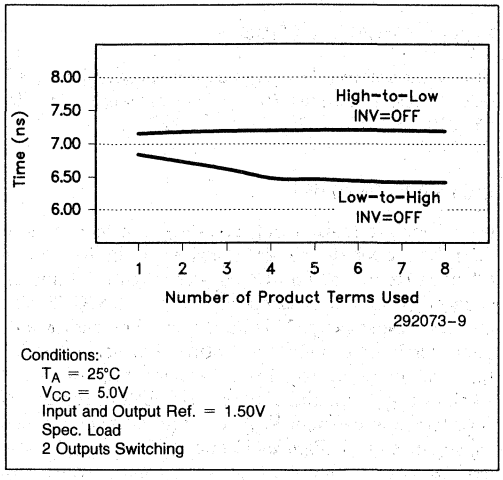


Figure 9. 85C220 t_{PD} vs Number of P-Terms Switching

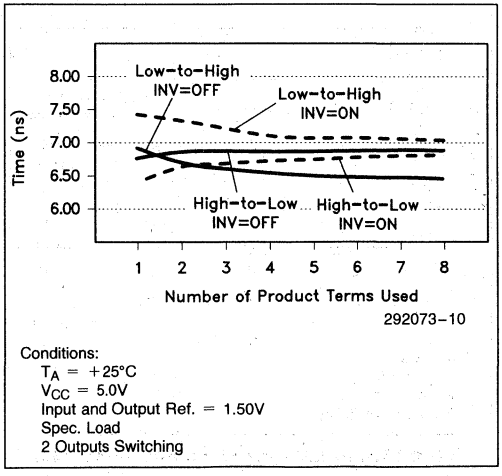


Figure 10. 85C224 t_{PD} vs Number of P-Terms Switching

t_{CO1} Characteristics

This section shows how application-specific factors affect Clock-to-Output (t_{CO1}) performance for the 85C220/85C224 PLDs.

t_{CO1} Characteristics

t_{CO1} (clock to register output valid) is in the 1.5 ns to 5.5 ns range for the 80 MHz device and in the 1.5 ns–6.5 ns range for the 66 MHz device. Figures 11 and 12 show t_{CO1} max. over temperature for the 85C220 and 85C224, respectively. This data was taken with 8 outputs switching at high temperature. For some designs, t_{CO1} minimum is also a key parameter. Figures 13 and 14 show t_{CO1} min. under worst case (fastest) conditions (low temperature, high V_{CC}, 1 output switching, probe capacitance only). Figures 15 and 16 show t_{CO1} for the devices with 1 to 8 outputs switching under worst case conditions (high temperature, low V_{CC}).

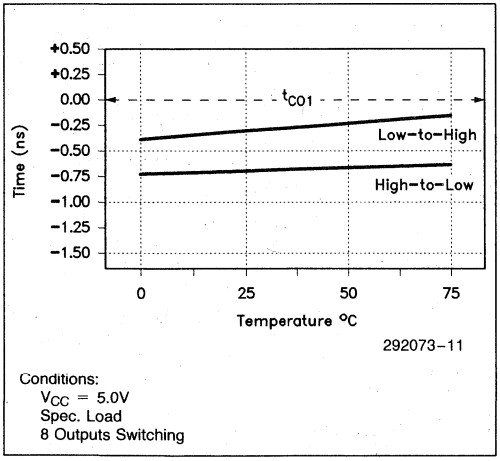


Figure 11. 85C220 t_{CO1} vs Temperature (Max.)

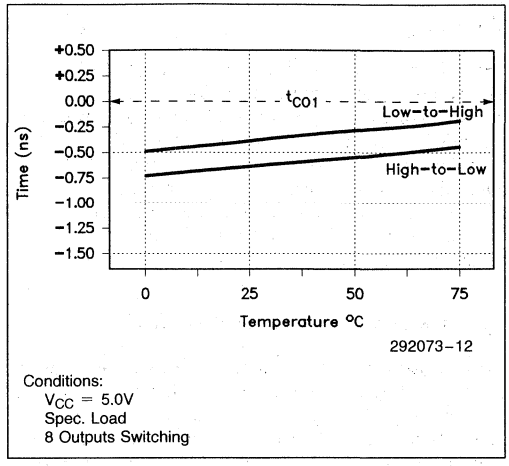


Figure 12. 85C224 t_{CO1} vs Temperature (Max.)

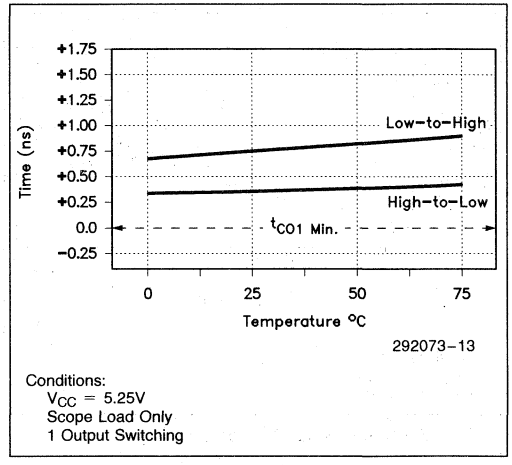
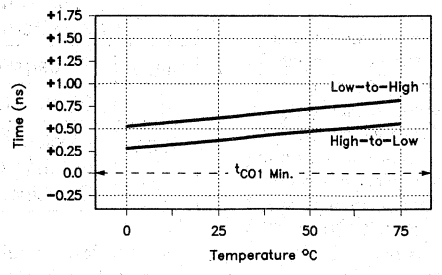


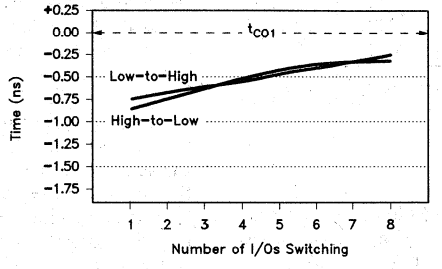
Figure 13. 85C220 t_{CO1} vs Temperature (Min.)



292073-14

Conditions:
V_{CC} = 5.25V
Scope Load Only
1 Output Switching

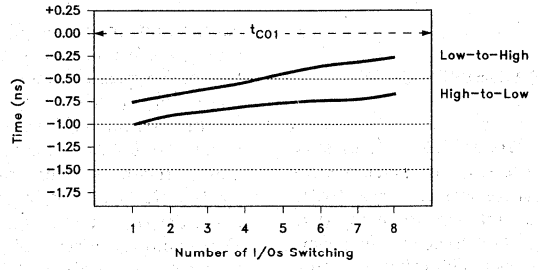
Figure 14. 85C224 t_{CO1} vs Temperature (Min.)



292073-15

Conditions:
T_A = +70°C
V_{CC} = 4.75V
Input and Output Ref. = 1.50V
Spec. Load

Figure 15. 85C220 t_{CO1} vs Number of Outputs Switching



292073-16

Conditions:
T_A = +70°C
V_{CC} = 4.75V
Input and Output Ref. = 1.50V
Spec. Load

Figure 16. 85C224 t_{CO1} vs Number of Outputs Switching

Register-to-Register Skew

In an ideal PLD, all registers clock simultaneously, with no skew between registers on the same device. In reality, clock signal routing and ground path differences cause some degree of skew between register outputs, but this does not pose a problem for most applications. When the skew is too great, however, or if the timing requirements of the application are especially stringent, output skew must be considered. Due to its high-speed double-metal process, output skew on 85C220/85C224 PLDs is very tight. Typical skew between fastest and slowest registers is shown in Table 2.

Table 2. 85C220/85C224 Register-to-Register Skew (Average)

Device	0°C		+70°C	
	High to Low (ps)	Low to High (ps)	High to Low (ps)	Low to High (ps)
85C220	120	310	120	340
85C224	120	280	120	310

I_{OL} Characteristics

Designers should note the relationship of I_{OL} for individual device outputs to the total I_{OL} for a device when designing with the 85C220/84C224. These devices can sink 64 mA per device (DC) while individual outputs can sink up to 12 mA of current. This is a steady-state current specification to prevent electromigration problems over the life of the device. The device maximum can be higher than 64 mA when duty cycle is taken into account. For example, in an 8-bit counter the duty cycle of the outputs is approximately 50%. For this application, the device maximum is 96 mA (8 outputs x 12 mA).

Tracking Critical Parameters over Temperature

Like all MOS devices, 85C220/85C224 devices slow down as the temperature increases. The difference between low and high temperature speeds is reflected in the minimum and maximum numbers for many data sheet specifications. It is important to be aware, however, of how these parameters track each other over temperature. For example, for the 85C220/85C224, some critical parameters are as follows:

80 MHz Device Parameter		Min (ns)	Max (ns)
t_{PD}	Propagation Delay	4	10
t_{CO1}	Clock to Output Valid	1.5	5.5
t_{CO2}	Clock to Output Valid (through Additional Combinatorial Output)	4.5	13

A surface level reading of the data sheet might lead designers to conclude that t_{PD} (Min. = 4 ns) will sometimes be faster than t_{CO1} (Max. = 5.5 ns). But in reality, the minimums in all three cases reflect the fastest possible speed, which occurs under optimum conditions (low temperature, 5.25V, 1 output switching, light load). The maximums reflect the slowest speed, which occurs at high temperature, 4.75V, 8 outputs switching, 30 pF load. Two of these factors, temperature and voltage, will always be the same for signals generated by the same device. Thus the three parameters will track each other closely over temperature and voltage under similar conditions. The number of outputs switching will have an affect on the speed, but will not change the relationship significantly (refer to "t_{CO1} vs. Number of Outputs Switching" earlier in this application note). Capacitive loading is the only factor that can change the way these parameters track each other. With light loading on t_{PD} and extremely heavy loading on t_{CO1}, it may be possible for t_{PD} to be faster than t_{CO1} at the system level.

Knowledge of how parameters track each other over temperature can help the designer to make intelligent judgements when performing system timing analysis.

Ground Bounce/Noise on Unswitched Outputs

As the performance of CMOS PLDs continues to increase, more attention must be paid to noise control. One of the most common noise problems is supply line noise. Supply line noise, often called ground bounce, can be a serious problem because high noise levels can affect the logical behavior of a circuit. A ground-bounce related problem often manifests itself in subtle ways, which makes debug diffcult. Noise problems are exacerbated by the fact that they are often intermittent. For example, depending on the number of outputs switching, an unswitched output might glitch high enough to create a false clock to a subsequent logic circuit or to cause memory or I/O devices to be deselected for a brief time, which can affect the integrity of data transfers.

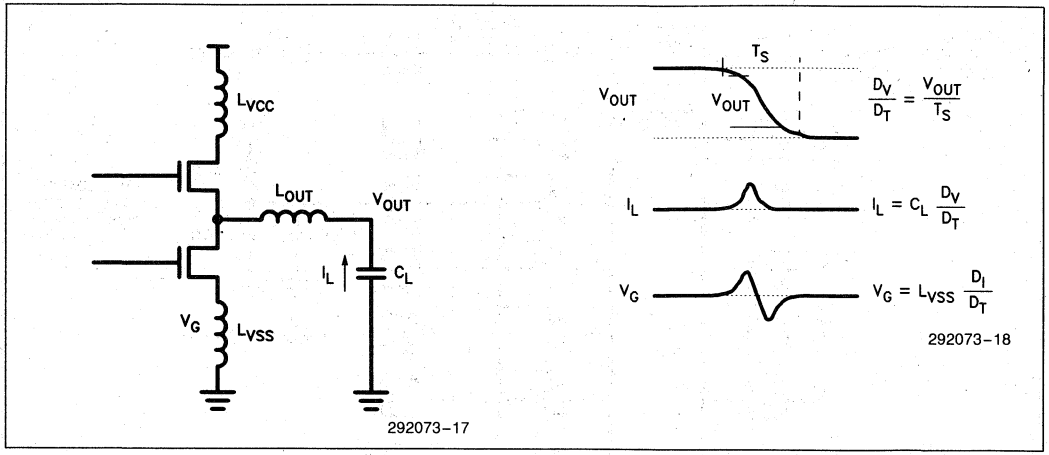


Figure 17. Basic Cause of Ground Bounce

Basic Cause of Ground Bounce

The fundamental cause of ground bounce is switching loaded output pins very quickly. Outputs are normally capacitively loaded, and there is always intrinsic inductance in device bond wires and package lead frames. The resulting structure is a classic LC circuit, as shown in Figure 17. Charging or discharging the circuit creates a damped oscillation. The magnitude, frequency, and duration of the oscillation are determined by the component values that make up the circuit.

V_{OUT} in Figure 17 is normally between 3V and 5V. Switching time (T_S) is in the range of 2 ns–4 ns. Load capacitance (C_L) is in the 30 pF–100 pF range. Inductance depends on the package design and materials, and is typically in the 7 nH–25 nH range. Based on these conditions, it is easy to see that ground bounce (V_G) can be a problem. Ground bounce further increases as more outputs switch simultaneously. In actual practice, the noise levels are less than the simple analysis shown in Figure 17 suggests. This is due to the self limiting nature of the output drivers. As the ground noise causes the local (internal) ground to rise, the drive of the device is reduced, which in turn reduces d_i/d_t.

Figure 18 shows the test setup for measurements. Measurements were made on 85C220 devices with 7 outputs switching simultaneously and the remaining output held high or low. The noise generated on the static output by the 7 outputs switching was then measured and recorded (worst case noise condition for a static output). All outputs have 22 pF of capacitance (>30 pF with scope probe), a 220Ω resistor to ground, and 330Ω resistor to V_{CC}. Decoupling is implemented with a 4.7 μF and a multi-layer 0.1 μF (ceramic) capacitor.

Figure 19 shows waveforms for the devices with 7 outputs switching. The unswitched outputs are held low. Waveforms are shown for the 85C220-80 (10 ns), 16V8A-10, 16L8-10, and 16L8-7 in Plastic DIP. An 85C220-80 and 16V8A-10 PLCC comparison is shown in Figure 20. Table 3 shows the relative inductance of the different packages.

The values shown represent a lumped load. In an actual system, the distributed load will reduce the values slightly.

Table 3. Package vs. Typical Inductance

Package Type	Typical Inductance
Plastic DIP (with Copper Leadframe)	11 nH
PLCC	6 nH

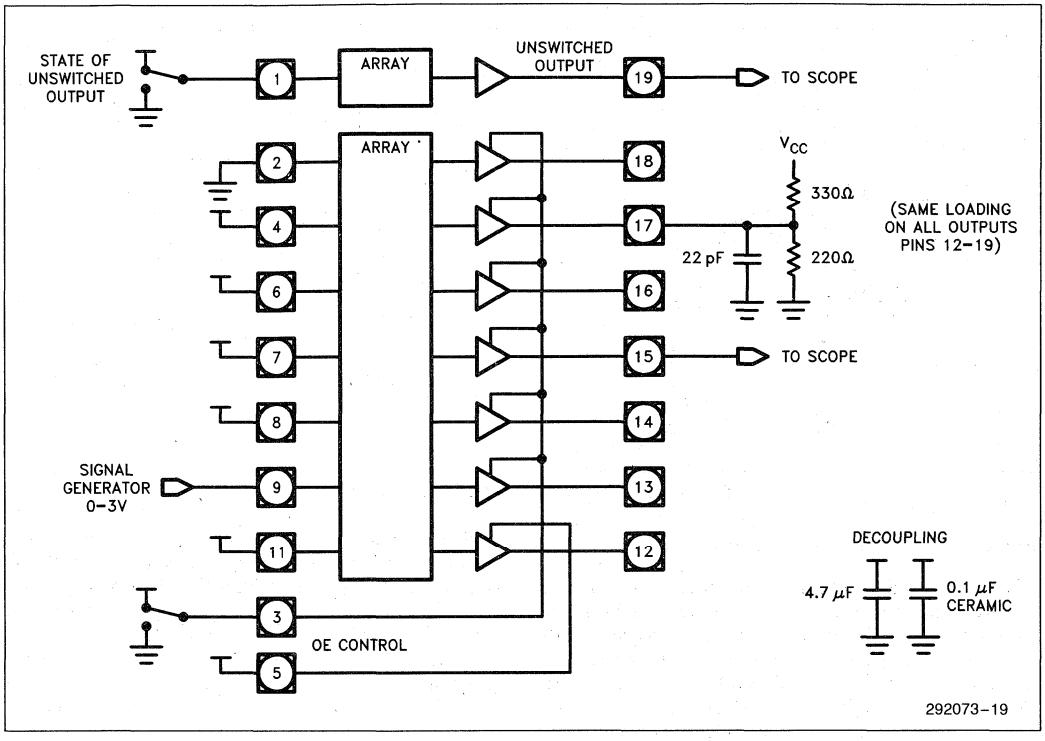


Figure 18. Ground Bounce Test Circuit

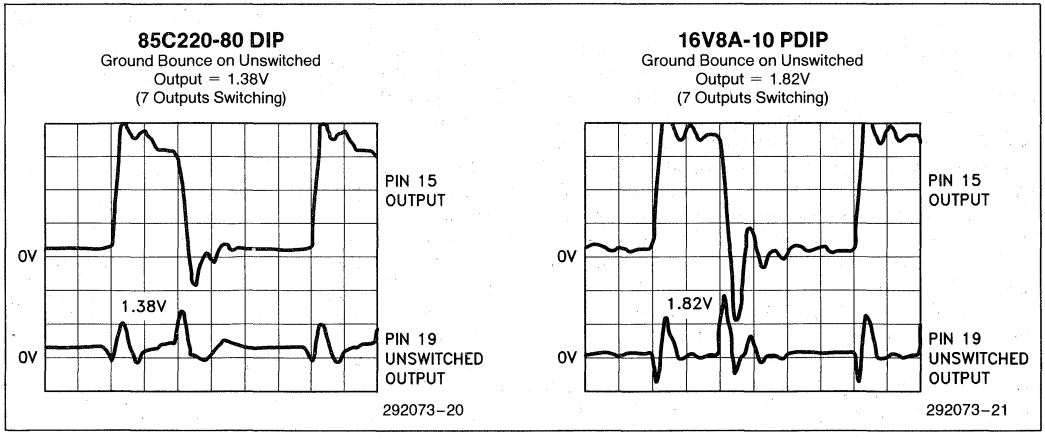


Figure 19. Ground Bounce (PDIP)—7 Outputs Switching

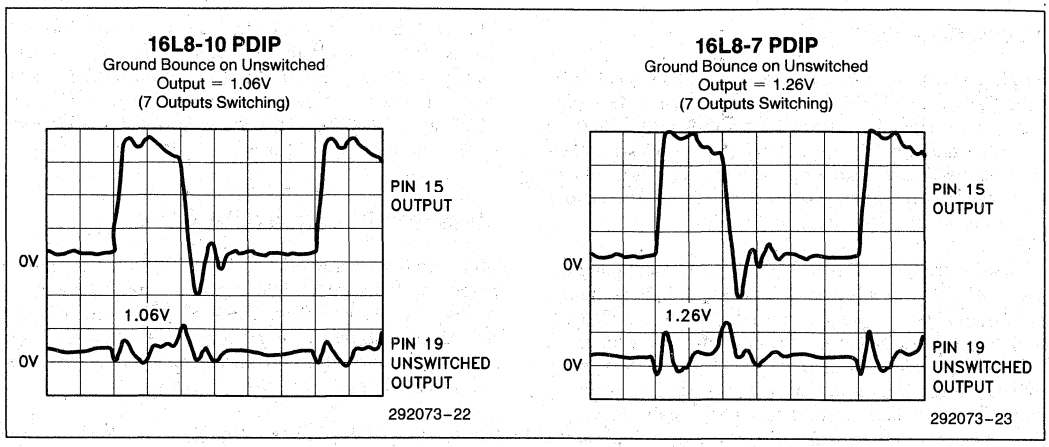


Figure 19. Ground Bounce (PDIP)—7 Outputs Switching (Continued)

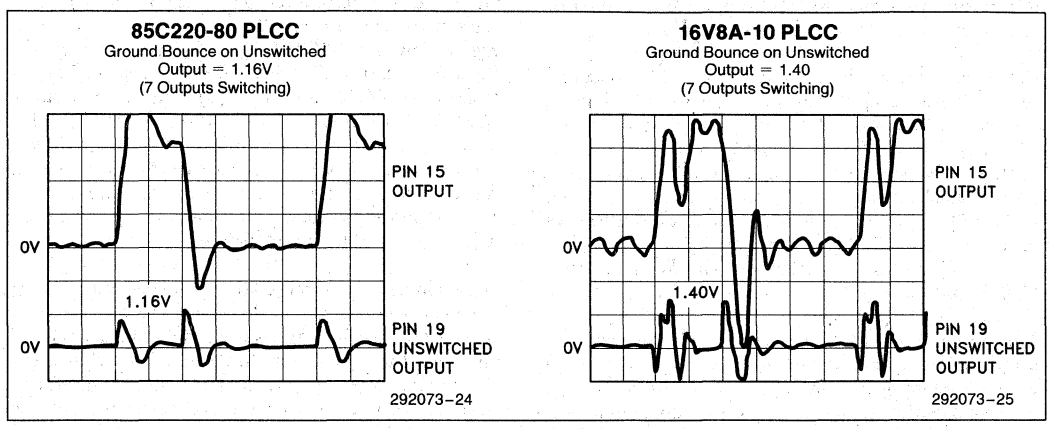


Figure 20. Ground Bounce (PLCC)—7 Outputs Switching

Design Considerations

High-performance CMOS devices such as the 85C220/85C224 are capable of driving large loads at fast edge rates. This capability means that noise control must be an important consideration during system design. Multi-layer PC boards utilizing ground and power planes provide low resistance and low inductance connections between the power sources and devices.

System noise control also requires good decoupling capacitance. Boards with power and ground planes but no decoupling capacitors can still have noise problems. High speed transients of devices may demand up to 500 mA of current, which can result in a volt or more of switching noise on the local supply lines. Decoupling capacitors can help prevent this performance degradation by providing a local power source during output transitions. With the addition of decoupling capacitors, it is possible to reduce the local supply noise to 200 mV or less.

Capacitor selection is important for this application since the frequencies involved in high-speed systems can exceed 100 MHz. High-frequency capacitors are necessary. The capacitors should provide low series inductance; leadless chip caps are the best choice, with short leaded capacitors as a more available second choice. The equivalent circuit for a capacitor is a series resonant circuit. If the inductive element in the capacitor is too high, the capacitor will appear inductive at high frequencies.

The unswitched output data shown in the previous figures provides a look at the noise level external to the device. Assuming that intelligent design practices have been followed to manage noise on the supply lines outside the device, internal noise should still be considered. The internal noise generated during switching transients is caused by output buffer design, package design, and output loading. Some suggestions for reducing noise are as follows:

- Select a low-inductance package such as PLCC.
- Reduce the output loading.
- Reduce the number of simultaneously switching outputs.
- Limit the voltage swing to 0V–3V by terminating outputs with resistors to ground.

Other Board Design Considerations

Designing high-speed P.C. boards requires closer attention to design issues that are not as important for slower systems. These elements include:

- Termination of transmission lines
- Clock signal routing
- Power distribution and heat dissipation

These topics exceed the scope of the application note. A discussion of them can be found in the "Design and Debugging" chapters of Intel's microprocessor hardware reference manuals. For example, refer to Chapter 11 of the *Intel386™ DX Hardware Reference Manual,* Order Number: 231732-004.

Metastability Characteristics

Metastability characteristics for PLDs are determined largely by the semiconductor process used in manufacturing the device. The Intel 85C220 and 85C224 μPLDs are fabricated on a superior 1.0 micron CMOS process that exhibits excellent metastability characteristics when compared to other processes.

Edge-triggered registers inevitably enter a metastable state when driven by data asynchronous to the register clock. Register resolution time is the time it takes for a register to resolve from this metastable state to a valid high or low. The 85C220/85C224 recover quickly from metastable states. Figure 21 shows register resolution time of the 85C220-80, along with comparative data on the 16V8A-10, 16R6-7, and 16R8-10 devices. These results are plotted in terms of Mean Time Before Failure (in years) vs. additional clock-to-output delay (in ns) for the clock and data rates shown. Note that the 85C220 performs much better than the bipolar devices, and is comparable to the 16V8A.

An example of how metastability affects system design is shown when trying to determine the maximum frequency possible for single-stage synchronization. In this situation, the register resolution time is combined with the maximum set-up and clock-to-output times for the device. The total is then used to determine both the maximum clock speed and the margins at lower speeds. Figure 22 shows maximum clock frequency for 100 year MTBF. The data rate is assumed to be 33% of the clock rate. Note that the 85C220 is the only PLD solution that provides margin at 50 MHz and that it provides higher margin than other devices at lower frequencies.

For a complete description of metastability characteristics and the methods used to obtain this information, refer to A-336, *Metastability Characteristics of Intel PLDs*, Order Number, 292071-001.

Acknowledgements

Many thanks to Gary Beckner, John Casey, Duane Chinnow, Liliyas Koumis, Ron Swartz, Mike Allen, and Chris Wawro for providing technical data and/or direction for this application note.

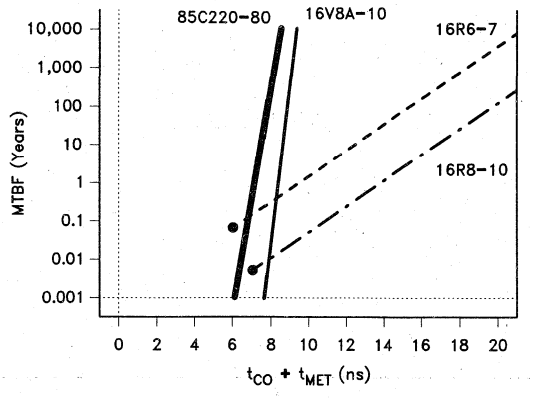


Figure 21. 85C220 Clock-to-Output Resolution vs MTBF

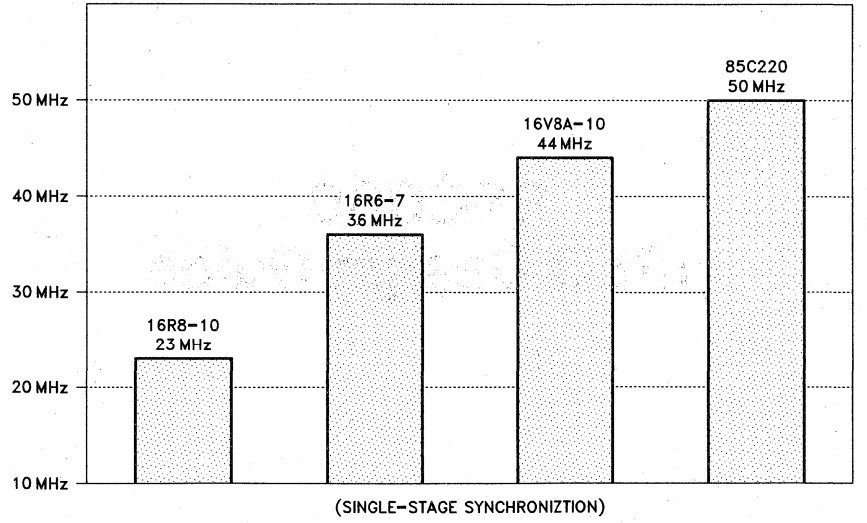


Figure 22. Maximum Frequency for 100 Year MTBF

intel®

APPLICATION NOTE

AP-339

October 1990

85C060 μPLD Design Guide

JOHN CASEY
PROGRAMMABLE LOGIC APPLICATIONS
INTEL CORPORATION

Order Number: 292074-001

85C060 μPLD DESIGN GUIDE

CONTENTS

CONTENTS

3

1.0 INTRODUCTION

Designers today are faced with the challenge of implementing very high performance systems in a cost-effective and timely manner. When these responsibilities are factored with concerns about reliability, power consumption, and migrating the design to the next performance level, the designer's job becomes more complex. When all these requirements are compared against the possible solutions spanning full-custom, semi-custom, application-specific, and programmable logic, the only sure result is a headache.

Intel's Microcomputer Programmable Logic Devices (μPLDs) can meet many of today's system requirements, as well as map the path to the next generation of products. Intel μPLDs provide a high-speed CMOS logic solution required by current microprocessors and VLSI peripherals.

DOCUMENT OVERVIEW

This design guide provides technical support for designers, design managers, components engineers, and others interested in using Intel's 85C060 μPLD. The information provided here is intended to support both the decision making process prior to the design and, the qualification process which occurs during and after the design. The format of this design guide is as follows:

Section 2—Product Overview: This section discusses the device highlights and architecture of the 85C060 μPLD.

Section 3—Specification Analysis: The key D.C. and A.C. Specs (from the data sheet) are discussed and compared against competitive devices. This section provides a baseline for comparison and device selection. Also, some insights are provided on how to best use the data sheet specifications.

Section 4—Advanced Design Issues: This section includes presentation of data and discussion of issues affecting high-speed systems designs. Topics include output slew rates, effects of capacitive loading of outputs, and synchronous/asynchronous register operation.

Section 5—Application Ideas: This section offers some ideas on how to use the unique performance/architecture combination offered by the Intel 85C060 μPLD.

Section 6—Programming/Development Support: This section offers details on existing Intel development/programming tools as well as third-party support. Also discussed is programming compatibility with other devices.

Section 7—Design Upgrade: There may be a desire to upgrade an existing design to the Intel 85C060. This task is discussed, specifically vis-a-vis the Intel 5C060, Altera/TI/EP6x0, 22V10, and 20RA10.

Related Documents

Title	Intel Order No
85C060 CHMOS μPLD Data Sheet	290246
85C090 CHMOS μPLD Data Sheet	290247
Intel Programmable Logic Handbook	296083
Metastability Characteristics of Intel EPLDs	292071
PLD Quality and Reliability Data Summary	293003

2.0 PRODUCT OVERVIEW

In programmable logic, several architectures have arisen as industry standards. Each architecture has a unique set of features. The Intel 85C060 μPLD represents the top performer in one of these industry standard architectures. The 85C060 is an upgrade of the existing Intel 5C060 and Altera EP600 devices. Although the architectures (i.e., pin-out, logic array, macrocell features, JEDEC map) are *exactly* the same, the 85C060 represents a very significant performance upgrade. It should also be noted that devices with this same architecture are manufactured by other leading programmable logic vendors including AMD and TI.

FEATURES

The features provided by the Intel 85C060 μPLD include the following highlights:

— 16 programmable macrocells (I/O pins)
— 24-pin DIP, 28-pin PLCC package
— EPROM cell, CMOS Technology (UV Erasable)
— 100% Silicon Testability
— Programmable Standby Current Mode (<100 μA)
— Low Operating Power (80 mA Max)
— High Performance Operation (12 ns t_{PD}, 7 ns t_{CO_1}, 66 MHz State Machine Frequency)
— Programmable Security Bit
— Programmable Register Type

PACKAGING

Figure 1 shows the pinouts of the DIP and PLCC packages for the 85C060 μPLD. The DIP version is available in plastic, one-time-programmable (OTP) and ceramic, which can be erased and reprogrammed. The PLCC version comes in a 28-pin OTP package.

PROCESS

The high performance capabilities of the 85C060 combine a standard architecture with Intel's advanced 1-micron CHMOS IIIE* EPROM technology. Also, extensive design work was accomplished on the input and output buffers to provide customers with a device that minimizes high-speed system related design issues including ground bounce, transmission line effects, and metastability.

DEVICE ARCHITECTURE

The overall device architecture, as presented in Figure 2, shows several architectural highlights of this device. In this 24/28 pin device there are 16 macrocells, each of which represents an I/O pin, buried register or dedicated input. There are also four dedicated input pins and two dedicated clock pins.

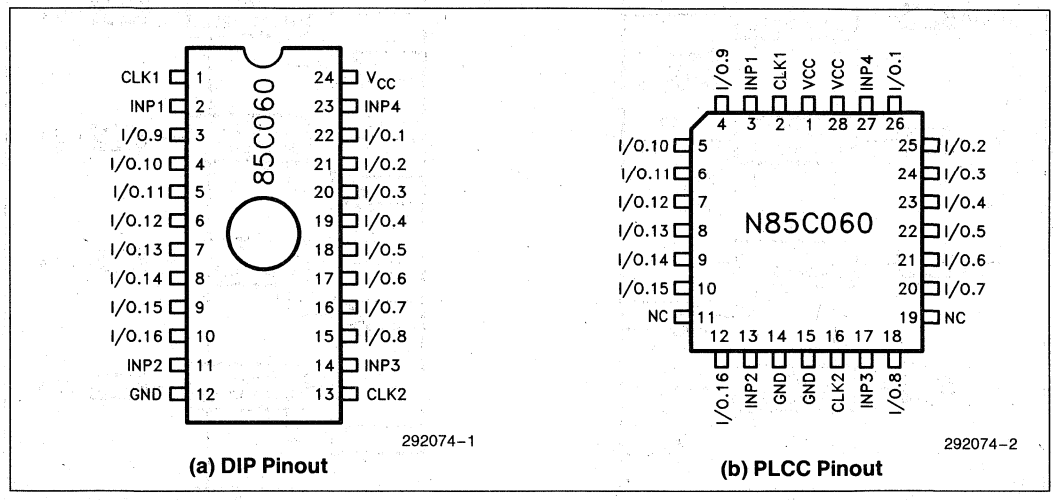


Figure 1. 85C060 μPLD Pinout Diagrams

*CHMOS is a patented process of Intel Corporation.

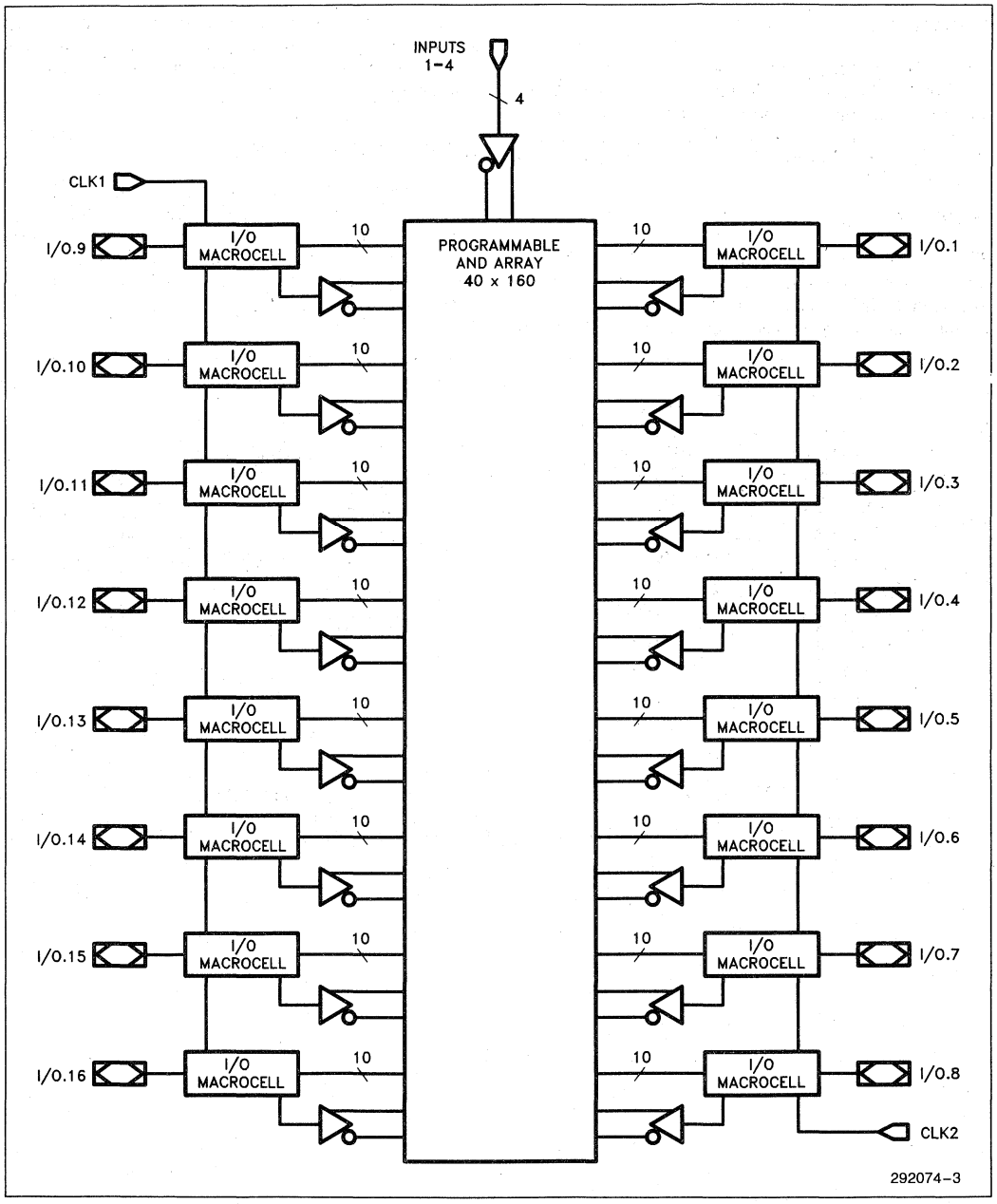


Figure 2. 85C060 Global Architecture

The dedicated clock pins offer the system designer added flexibility. The CLK1 pin provides a synchronous clock input to macrocells 9–16 and the CLK2 pin provides a dedicated synchronous clock input for macrocells 1–8. The advantages of this architecture feature are quite obvious as the list of applications could include:

a) Dual state-machine operation within the same device

b) Using logic that requires operations at CLK_x and $\overline{CLK_x}$

c) Implementing system control logic requiring both a 1x CLK and a 2x CLK (as can often be the case with the Intel 80386DX/SX and 80286 microprocessors)

d) Use of some macrocells as "input latches" accepting asynchronous inputs, whose states are subsequently acted upon using another clock input.

MACROCELL ARCHITECTURE

The next level of detail to examine the 85C060 is at the macrocell level. Figure 3 provides an overview of the macrocell architecture. There are several key features that make this device very useful to system designers by overcoming limitations of the standard PAL*/GAL**/22V10 capabilities. Every macrocell of the 85C060 has 8 sum of product terms for standard logic implementation. There is a separate product term for Asynchronous Register clear (Reset) for each register. In addition, there is a product term that can be programmed as either an asynchronous clock input to the register or as an output enable control. This asynchronous clocking capability for each macrocell provides yet another level of clocking flexibility to an already advanced architecture. The output enable control available on each macrocell allows any I/O pin to be tri-stated and provides for full control of pins used as both inputs and outputs. The feedback from each macrocell can be direct register feedback or pin feedback. Several macrocell options are detailed in the 85C060 Data Sheet (Lit. No. 290246). Each macrocell can be configured to implement registered or combinational logic. Register types available include D, T, JK and RS. This provides designers register types not found in many of the common programmable logic devices. Register type selection is also a common way to reduce product term requirements (rather than just going to higher product term devices), as logic development tools can minimize equations for the best fit. Whether registered or combinational logic is implemented, there is also an inversion control within each macrocell. Again, this can help designers minimize logic and control output polarities by using DeMorgan's inversion rules on equations.

3

*PAL® is a registered trademark of Advanced Micro Devices.
**GAL® is a registered trademark of Lattice Semiconductor, Inc.

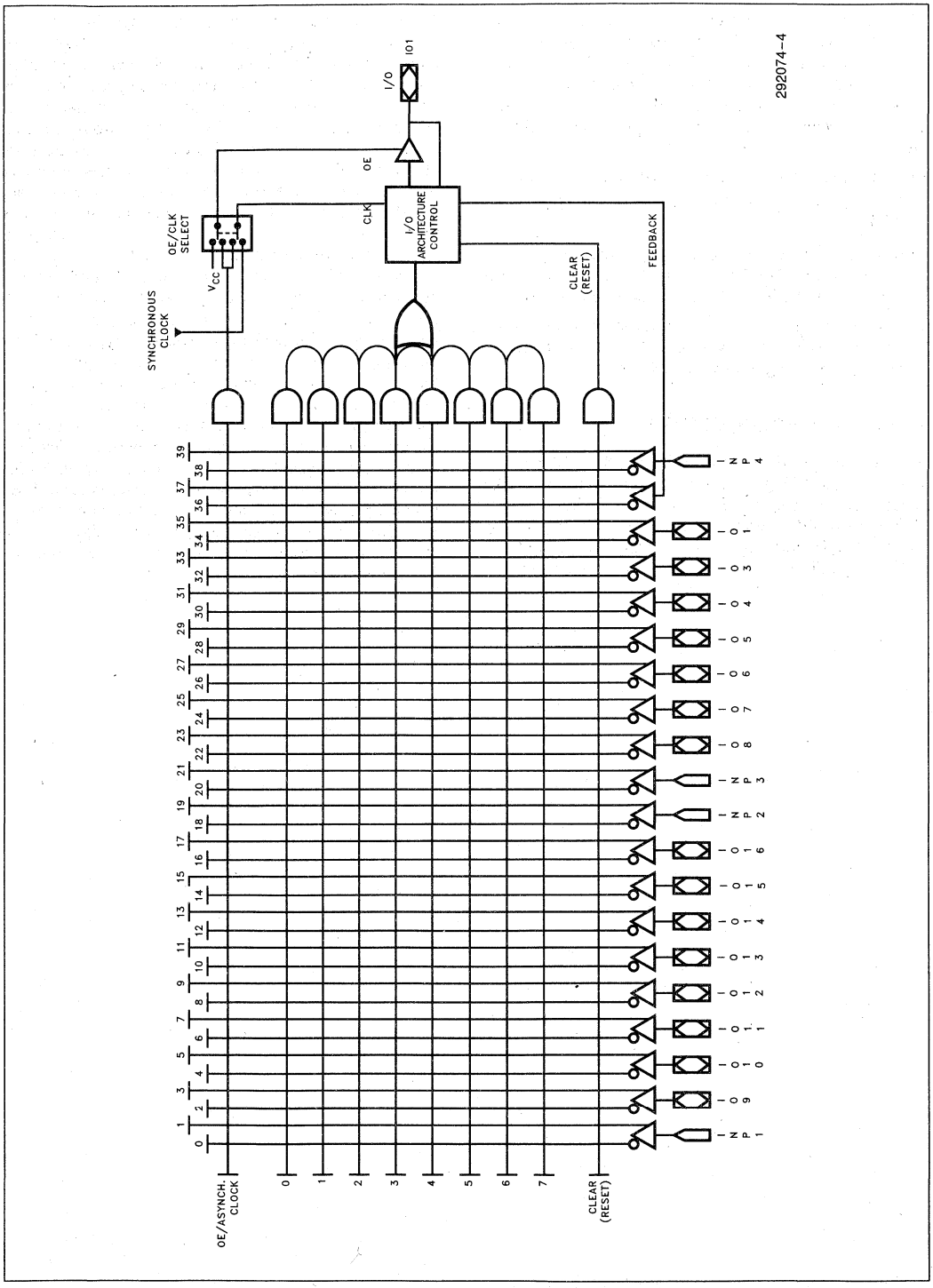


Figure 3. 85C060 Macrocell Architecture

3.0 SPECIFICATION ANALYSIS

When designing with any IC, a designer would like to know two things:

1) How is the part guaranteed to act (i.e., what are its specifications)

2) How will it act in it's actual design environment

The first item above will be covered in this section. Using the data sheet A.C. and D.C. specifications a designer can perform a "paper analysis" of the circuit. The questions to be answered here are "How does the Intel 85C060 μPLD compare with competitive devices" and "How can the A.C. and D.C. specs best be used?"

The second item is much harder to quantify due to the vast differences between designs. However, a great deal of bench data has been collected and presented in Section 4, which will help the designer make these assessments.

D.C. characteristics of the Intel 85C060 μPLD are covered first. Next follows a comparison of these values with competitive devices. Following this is an overview of the A.C. specifications for the 85C060 and, finally, a comparison of these specifications with other devices.

3.1 D.C. Characteristics Analysis

TTL Compatibility

D.C. characteristics address the direct current components of operating any IC. The areas covered include input and output voltage and current levels, and I_{CC} supply current consumption.

The input voltage levels (V_{IH} and V_{IL}) show that the 85C060 can be driven by standard TTL or CMOS logic components, so the designer doesn't need to worry about mixing logic families—even though the 85C060 is a CMOS PLD. The V_{OH} and V_{OL} show that the 85C060 can drive TTL loads over the specified test conditions. CMOS loads are capacitive in nature and do not require much current (typically measured in μA).

I_{OL} Specifications

The V_{OL} test condition shows that each 85C060 is capable of handling a 12 mA load while maintaining the output at or below 0.45V. Refer to Figure 17 (in Section 4) if another V_{OL}/V_{OH} value is required. That figure provides typical values for the output driver of the 85C060.

Also in the D.C. specifications of the 85C060 is a note specifying the total I_{OL} for each "bank" of 8 macrocells is 64 mA. This is the maximum recommended D.C. (steady state) current for this device. Even though the 85C060 can sink 12 mA on each output, this 64 mA/bank limit should be observed to reduce electromigration effects. The duty cycle of each output should be incorporated when calculating the steady state device current. For example, each output of a 4-bit counter is active 50% of the time. The maximum current for each bank is 96 mA.

The input and output leakage current specifications are ± 10 μA (per pin). This is an average value for CMOS devices. Typically, though, the total leakage for all pins (combined) will be within this ± 10 μA range.

I_{CC} Specifications

Probably the D.C. specification of the greatest concern to most designers is the I_{CC} value. This value tells the designer how much current will be required and how much heat (watts) will be generated by each device. Therefore, the I_{CC} specifications of a device will affect both the power supply requirements and the board reliability. To generate I_{CC} values, the device is tested with no load to find out how much power is consumed by the device itself.

Since CMOS devices consume most of their power during transitions, I_{CC} will greatly vary with the frequency of the clock or other inputs. For this reason not only are two values of I_{CC} specified in the D.C. characteristics (at 1 MHz and 66 MHz), but an I_{CC} versus frequency graph is given at the back of the data sheet allowing more precise current/power calculations. This graph provides "typical" I_{CC} values (i.e., V_{CC} = 5.0V, temp = 25°C).

Another specification related to I_{CC} is the standby current specification, I_{SB}. This specification gives both typical and maximum values for power consumption when the 85C060 is in standby mode.

To be in standby mode the device must be programmed with TURBO = OFF and no inputs (except clock inputs) can have changed state for 75 ns. Wakeup time adds an additional 25 ns to the propagation delay through the device as measured from the first input change (see Figure 4). The standby mode programming option provides designers of power-sensitive applications such as laptop PCs with the capability to greatly reduce the system's power budget.

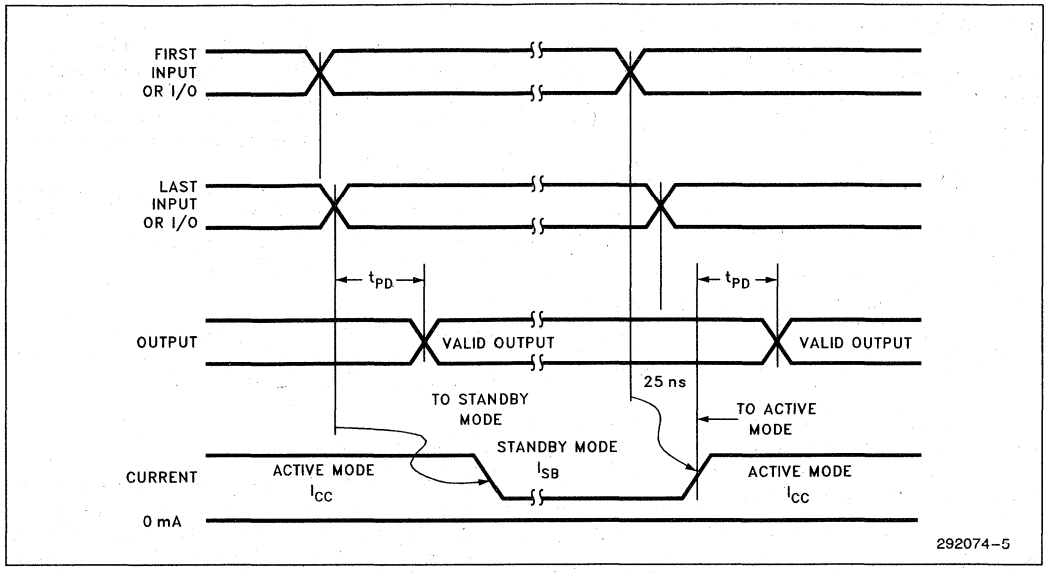


Figure 4. 85C060 Standby and Active Mode Transitions

3.2 D.C. Specification Comparison

The voltage specifications for the 85C060 and competitive devices do not differentiate where these devices can be used. For this reason Table 1, below, concentrates on the current specifications.

The conclusions from this table are clear. The Intel 85C060 provides the lowest power consumption (I_{CC}) while providing the highest output drive capability (12 mA). Also, it's values for standby current are comparable to the other devices (although the AMD PALCE630 does not offer a standby mode).

Table 1. D.C. Specification Comparison

Spec	I_{CC} (@ 1 MHz)		I_{SB}		Output Current	
Part No.	**Typical**	**Max**	**Typ**	**Max**	I_{OL}	I_{OH}
Intel 85C060	3 mA	8 mA	20 μA	100 μA	12 mA	−4 mA
Intel 5C060	10 mA	15 mA	20 μA	100 μA	4 mA	−4 mA
Altera EP630	5 mA	10 mA	20 μA	150 μA	4 mA	−4 mA
Altera EP610	3 mA	10 mA	20 μA	100 μA	4 mA	−4 mA
TI EP610	3 mA	10 mA	20 μA	100 μA	4 mA	−4 mA
AMD PALCE630	Not Availl.	Not Avail.	Not Supported	Not Supported	8 mA	−4 mA

3.3 A.C. Specification Analysis

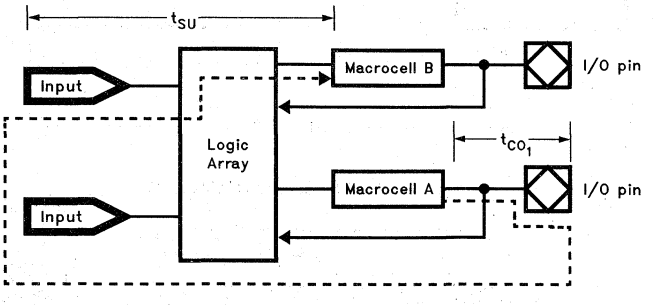
COMBINATORIAL

A.C. specifications provide the real "meat" of the data sheet by stating the guaranteed performance of the device. The first section of A.C. characteristics are the combinatorial mode A.C. characteristics. These state timings of non-registered logic paths in the 85C060, including propagation delays (t_{PD1} and t_{PD2}), output enable/disable times (t_{PZX} and t_{PXZ}), and asynchronous reset time for any register (t_{CLR}). These specifications are stating the times from any I/O or input pin until the selected function is performed. Propagation delays include AND/OR logic provided by up to eight product terms.

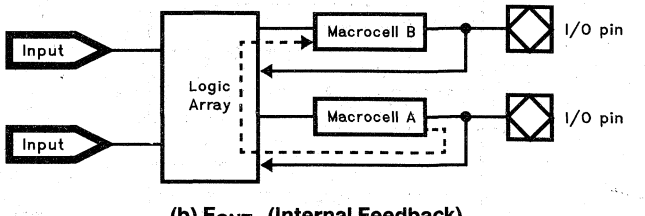
SYNCHRONOUS

The next table in the 85C060 data sheet is the "Register Mode—Synchronous Clock A.C. Characteristics" which includes specifications related to using any macrocell in registered mode which is clocked with CLK1 or CLK2. The specifications of primary importance here revolve around the register setup time (t_{SU}) and register clock to output time (t_{CO_1}). The setup time determines how long before the clock edge arrives that valid data must already be at the input pin(s). The clock to output time states the delay from the time data is clocked at the register until valid data shows up at the output pin. The reason these two specifications are of concern is that together they determine the worst case throughput performance in registered mode. One highlight of the 85C060 is how closely the t_{SU} and t_{CO_1} specifications are to standard "E-PAL" specifications. Section 3.4 includes a summary of 85C060 versus PAL performance. The maximum counter frequency, F_{CNT_1}, represents the maximum counter frequency with the 85C060 using external feedback. External feedback means routing an output pin to another pin as an input. In this case the designer incurs both input and output buffer delays—but, this also simulates the activity of a multiple device counter or state machine. A related specification, F_{CNT_2}, represents the maximum counter (state machine) frequency using internal device feedback. F_{CNT_2} is higher than F_{CNT_1} due to elimination of an input and output buffer from the feedback path. Figure 5 provides an overview of both F_{CNT} values.

The designer must also be aware that the F_{MAX} specification states the maximum frequency at which the registers in the 85C060 can be clocked. This is due to the physical limitations on the period of the clock input (it could also be limited by register setup time). Therefore, $F_{MAX} = 1/t_{CW}$ (= 1/min clock width).



(a) F_{CNT_1} (External Feedback) $= \frac{1}{t_{SU} + t_{CO_1}}$



(b) F_{CNT_2} (Internal Feedback)

Figure 5. F_{CNT} Overview

Designs utilizing programmable logic can make full use of the 85C060 by understanding the applications of F_{CNT_1} (used for multiple device state machines), F_{CNT_2} (used for smaller, single device state machines), and F_{MAX} (used for register pipelining applications).

Another A.C. specification of interest is t_{CO_2} which represents the time from CLK high to output valid fed through a combinatorial macrocell. Figure 6 depicts the difference between t_{CO_1} and t_{CO_2}. The reason t_{CO_2} is specified to show the advantages of using the internal feedback capabilities of Intel PLDs. Similar to the advantage gained in counter frequency between F_{CNT_1} and F_{CNT_2}, t_{CO_2} offers combinational logic speeds comparable to traditional PAL solutions by decreasing the number of input and output buffer delays encounter.

ASYNCHRONOUS

For each of the synchronous clock A.C. characteristics there is an asynchronous clock A.C. characteristic specified in the data sheet. Each macrocell of the 85C060 can be programmed to be clocked with an asynchronous clock (generated by a separate product term in each macrocell). The asynchronous clock values closely mirror the synchronous specifications in both definition and value, but, there are a few notable differences.

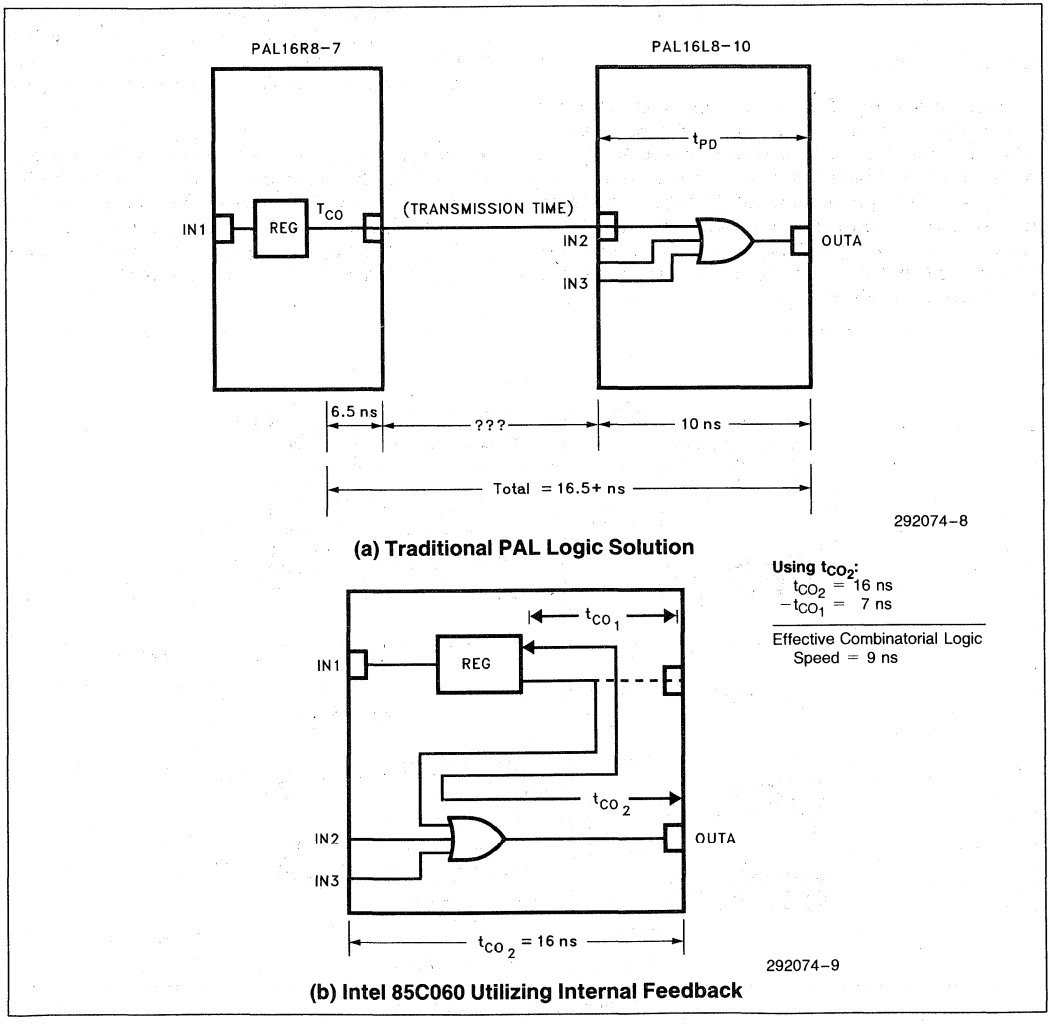


Figure 6. Equivalent Combination Logic Speed Using t_{CO_2}

The synchronous register setup and hold time specifications (t_{SU} and t_H) are 8 ns and 0 ns, respectively. The asynchronous register specifications show a shifting of specifications due to differences in internal paths. The asychronous register setup and hold time specifications (t_{ASU} and t_{AH}) are 3 ns and 5 ns, respectively. A similar shifting is seen if register setup and clock-to-output specifications are compared for synchronous and asynchronous operation as detailed in Figure 7.

Understanding of these differences between synchronous and asynchronous specifications can be useful to a designer. If a system design requires a short setup time (in the 3 ns–7 ns range) one or more of the macrocells of the 85C060 could be used in the asynchronous clocking mode. Either a control input or the system's synchronous clock could be routed to an input or I/O pin to establish the necessary "asynchronous" clock. With this technique, setup times as low as 3 ns can be met by the 85C060, although there is the trade off of a longer register clock-to-output delay. Low setup times are often required to accommodate output valid delay specifications of many microprocessors and system peripherals.

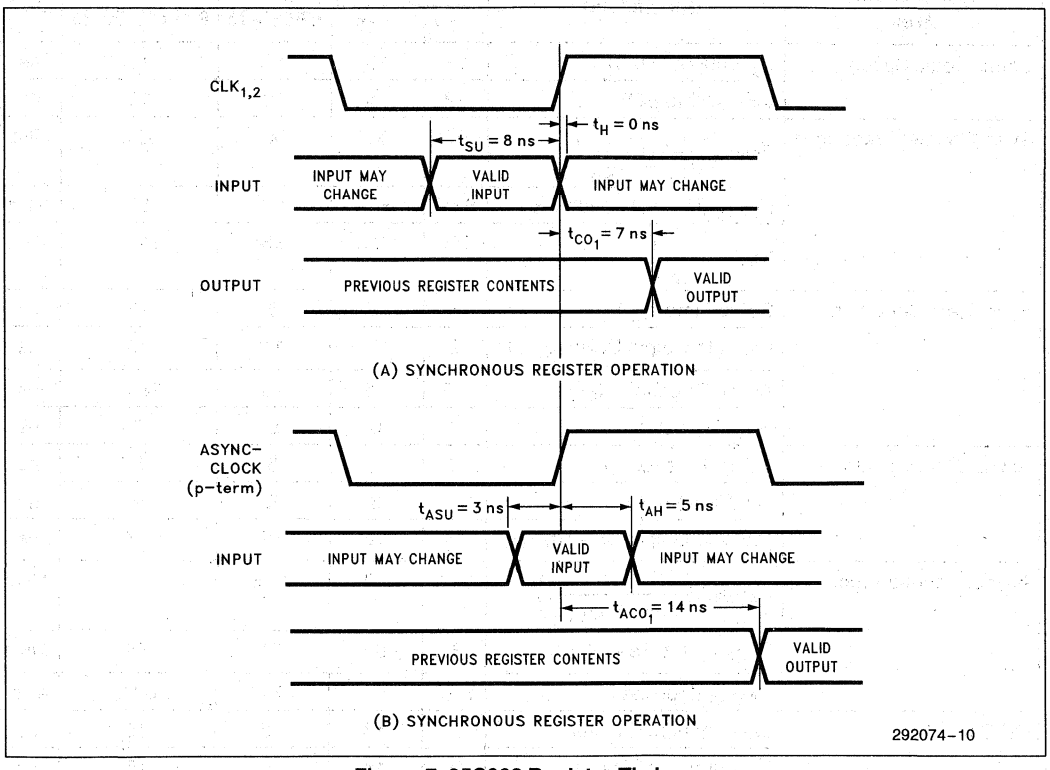


Figure 7. 85C060 Register Timings

3.4 A.C. Specifications Comparison

The comparison made here involves other devices with the same architecture as the Intel 85C060. This includes devices from Altera, TI and AMD. It should be noted that Tables 2 and 3 (below) do not compare all A.C. specifications, rather only those best reflecting the performance of these devices. Even though the architectures of these devices (i.e., pinout, product term, macrocell configurations, etc.) are all exactly the same there are some differences in the test conditions used to generate A.C. specification. These differences are noted.

The first comparison involves the higher speed parts—those in the 12 ns–15 ns range for t_{PD}. The second comparison will involve slower parts—those with a t_{PD} of 25 ns. The final comparison shows a summary of 85C060 performance versus standard PAL devices.

Table 2. A.C. Specification Comparison—High Performance

Specification Area	Specification	Intel 85C060		Altera EP630-15	AMD PALCE630-15	Unit
		−12	−15			
Comb. Logic Speed	t_{PD1} (Input to I/O)	12	15	15	15	ns
	t_{PD2} (I/O to I/O)	12	15	16	15	ns
Sync. Counter Frequency	$F_{CNT_1}\left(\frac{1}{t_{SU} + t_{CO_1}}\right)$	66	50	50	45.5	MHz
	F_{CNT_2} (Internal Feedback)	83.3	66	83	50	MHz
	$F_{MAX}\left(\frac{1}{t_{CW}}\right)$	100	83.3	83	62.5	MHz
Sync. Reg. Operation	Setup Time (t_{SU})	8	12	9	12	ns
	Clock to Output Delay (t_{CO_1})	7	8	11	10	ns
	Async Reg Clear (t_{CLR})	15	15	15	N/A	ns
	Hold Time (t_H)	0	0	0	0	ns
Async. Reg. Operation	Setup Time (t_{ASU})	3	4	6	5	ns
	Hold Time (t_{AH})	5	6	6	9	ns
	Clock to Output Delay (t_{ACO_1})	14	16	15	17	ns
Async. Counter Freq.	$F_{ACNT_1}\left(\frac{1}{t_{ASU} + t_{ACO_1}}\right)$	58.8	50	47.6	45	MHz
	F_{ACNT_2} (Int. Feedback)	83.3	66	71	48	MHz
	F_{AMAX} (Pipelined)	83.3	66	71	62.5	MHz
OE Control	t_{PZX} (Input to Enable)	15	18	15	15	ns
	t_{PXZ} (Input to Disable)	15	18	15	15	ns

Table 3. A.C. Specification Comparison—Mid Range Device Propagation Delay = 25 ns

Specification Area	Specification	Intel 85C060-25	Altera/TI EP610-25	AMD PALCE630-25	Unit
Comb. Logic Speed	t_{PD1} (Input to I/O)	25	25	25	ns
	t_{PD2} (I/O to I/O)	25	27	25	ns
	t_{CLR} (Async Reg Clear)	25	27	N/A	ns
Sync. Register Performance	$F_{CNT_1}\left(\frac{1}{t_{SU} + t_{CO_1}}\right)$	40	27.8	37	MHz
	F_{CNT_2} (Int Feedback)	40	40	40	MHz
	$F_{MAX}\left(\text{Pipelined} = \frac{1}{t_{CW}}\right)$	50	47.6	50	MHz
Sync Register Operation	t_{SU} (Setup Time)	15	21	15	ns
	t_H (Hold Time)	0	0	0	ns
	t_{CO_1} (Clock to Output Delay)	10	15	12	ns
Async Register Performance	F_{ACNT_1} (Ext. Feedback)	33.3	28.5	28.6	MHz
	F_{ACNT_2} (Int. Feedback)	40	40	29	MHz
	F_{AMAX} (Pipelined)	50	47.6	41.6	MHz
Async Register Operation	t_{ASU} (Setup Time)	5	8	8	ns
	t_{AH} (Hold Time)	8	12	12	ns
	t_{ACO_1} (Clock to Output Delay)	25	27	27	ns
OE Control	t_{PZX} (Input to Enable)	25	25	25	ns
	t_{PXZ} (Input to Disable)	25	25	25	ns

At the "slower" end of the product line (t_{PD} = 25 ns), the 85C060 provides an upgrade to existing 25 ns t_{PD} devices—especially in the area of synchronous register operations. From there a designer can migrate a design to the 85C060-15 and finally to the state-of-the-art in architecture/performance, the Intel 85C060-12 μPLD.

PAL Comparison

A common application for higher integration devices such as the 85C060 μPLD is upgrading existing PAL designs. Table 4 summarizes the performance specifications of the 85C060 and standard PAL devices. The primary register-related specifications of the 85C060, t_{SU} and t_{CO_1}, are very close to E-PAL performance levels. This results in F_{CNT_1} specifications between the two devices which are very close. Also of note is that F_{MAX} of both are at 100 MHz. The major differences between the 85C060 and the PAL devices are in the t_{PD} and I_{CC} values. The 85C060 cannot meet the faster combinational logic speeds of PAL devices, however, PALs require about 3 times more current than the 85C060 μPLD. This I_{CC} difference is important in many design areas including power supply sizing, cooling requirements, and overall system reliability.

Table 4. 85C060 μPLD vs PAL Performance Summary

Specification	85C060	"E-PAL" 20xx-7	"D-PAL" 20xx-10	Units
t_{SU}	8	7	10	ns
t_{CO_1}	7	6.5	8	ns
F_{CNT_1}	66	74	55.5	MHz
F_{MAX}	100	100	71.4	MHz
t_{PD}	12	7.5	10	ns
I_{CC}(max)	80	210	210	mA

4.0 ADVANCED DESIGN ISSUES

As system designs climb to higher speeds and time to market becomes more critical, designers require more detailed information than is available in the data sheet. The main emphasis here is to provide designers with characterization data of key aspects of the 85C060 μPLD. This data will help avoid unforseen problems in the design, prototype, and production phases.

Data was measured with the output load specified in the 85C060 data sheet (shown also in Figure 8), unless otherwise specified. The topics covered in this section are:

- Output Slew Rates
- Combinational Logic Concerns (t_{PD} Characteristics)
- Synchronous Register Operation Characteristics
- Asynchronous Register Operation Characteristics
- Output Current Characteristics
- Design Considerations

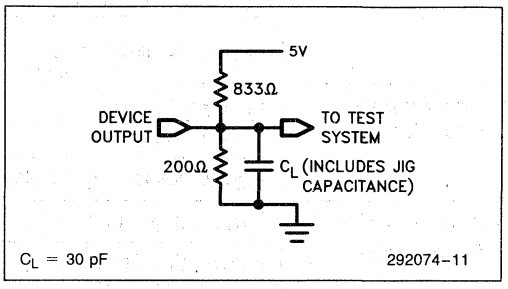
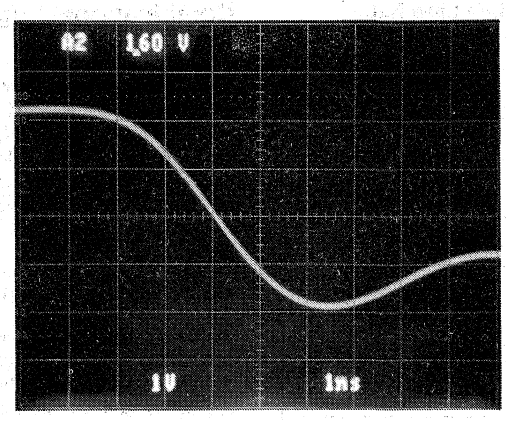


Figure 8. A.C. Testing Load Circuit

Output Slew Rates

Several key advantages of using Intel μPLDs are derivatives of the output buffer design. One of these is slow output slew rates which minimizes system noise. Signal ringing (noise) can result if output rise times are shorter then twice the signal transmission time. Thus, the faster the edge rate of any device output the shorter the board trace that is allowable before transmission line effects must be considered. This generally involves series or parallel termination and more consideration of device location and signal routing. The 85C060 μPLD seeks to limit these design problems by providing lower output slew (edge) rates.

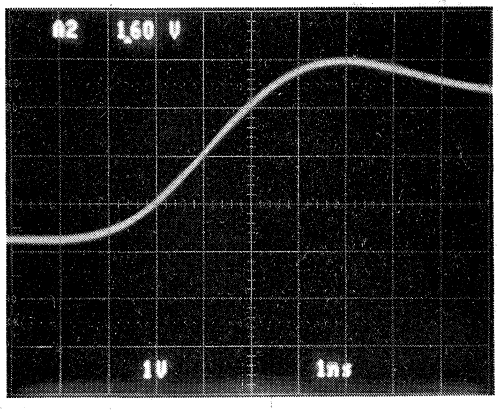
Figures 9 and 10 provide a sample of the 85C060 output slew rate characterization. Table 5 shows there is very little variation over temperature, although the values do decrease as temperature increases. Table 6 shows there is very little variation over the number of outputs switching simultaneously, although it does decrease as the number of outputs increases. The results show output slew rates comparable to those of bipolar PALs and much lower than CMOS GALs (which are in the 3–5 V/ns range). The edge rates provided by the 85C060 provide designers with flexibility without constantly worrying about transmission line effects.



292074-12

Temp. = 25°C
Spec. Load
V_{CC} = 5.0V
8 Outputs Switching

Figure 9. 85C060 Output Slew—H → L Transition



292074-13

Temp. = 25°C
Spec. Load
V_{CC} = 5.0V
8 Outputs Switching

Figure 10. 85C060 Output Slew—L → H Transition

Table 5. 85C060 Output Slew Rate vs Temperature

Transition	Package Type	Temp (°C)	Slew Rate (V/ns)
L → H	PDIP	0	1.1
	PDIP	25	1.1
	PDIP	70	1.0
	CerDIP	0	1.0
	CerDIP	25	1.0
	CerDIP	70	0.9
H → L	PDIP	0	1.2
	PDIP	25	1.2
	PDIP	70	1.1
	CerDIP	0	1.2
	CerDIP	25	1.2
	CerDIP	70	1.1

V_{CC} = 5.0V
No outputs switching = 8

Table 6. 85C060 Output Slew Rate vs Number of Outputs Switching

Transition	Package Type	No. of Outputs Switching	Slew Rate (V/ns)
L → H	PDIP	2	1.1
	PDIP	8	1.1
	PDIP	15	1.0
	CerDIP	2	1.1
	CerDIP	8	1.0
	CerDIP	15	0.9
H → L	PDIP	2	1.3
	PDIP	8	1.2
	PDIP	15	1.2
	CerDIP	2	1.3
	CerDIP	8	1.2
	CerDIP	15	1.1

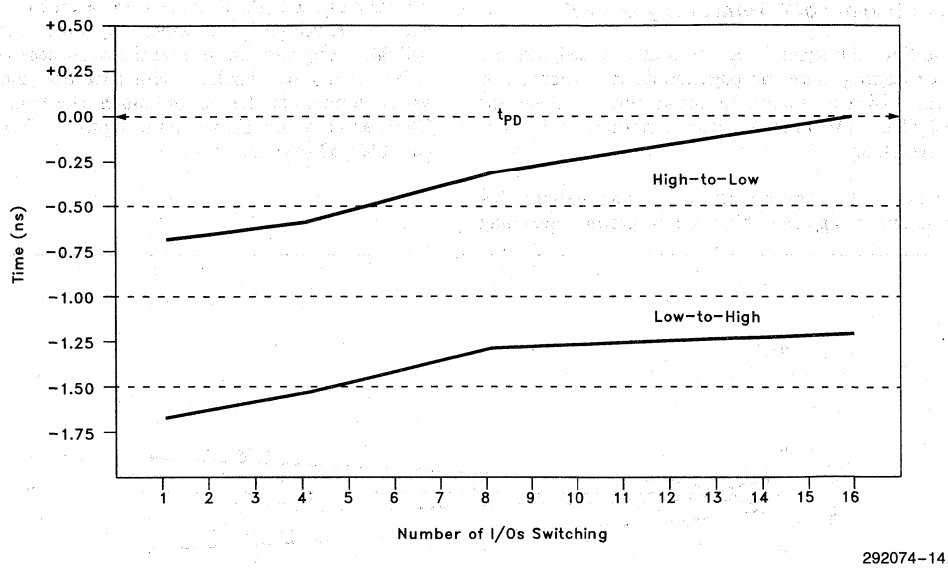
V_{CC} = 5.0V
Temp. = 25°C

Combinational Logic Performance (t_{PD})

This section shows how propagation delay for the 85C060 μPLD is affected by factors that vary from one application to another. Note typical parts have propagation delays 1–2 ns below the value specified in the data sheet. Data such as this can aid designers required to "fine tune" their designs and/or provide worst-case timing analyses.

t_{PD} vs Number of Outputs Switching

As the number of device outputs switching simultaneously increases, average propagation delay through devices also increases. This increase is related to package power and ground leads to channel the additional current. Figure 11 shows the relation of t_{PD} to the number of outputs switching. Note, this data reflects worst case conditions (high temperature, low V_{CC}) and all outputs have loads as specified in the 85C060 data sheet.

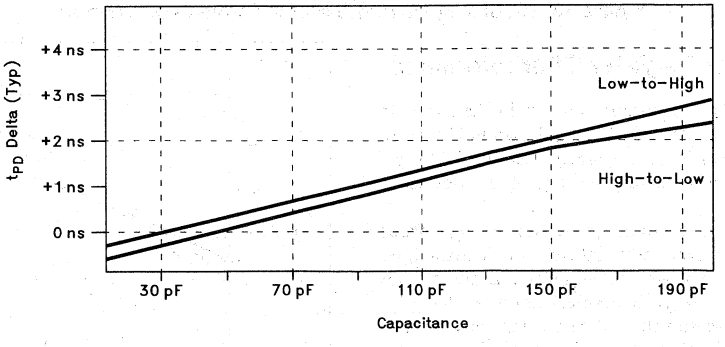


Conditions:
T_A = 70°C
V_{CC} = 4.75V
C_L = 30 pF

Figure 11. 85C060 t_{PD} vs Number of Outputs Switching

t_{PD} vs C_L

Knowledge of how devices behave as capacitive loading is increased is an important consideration when designing high-speed systems. Figure 12 shows derating from specified values for a typical 85C060 at high temperature, low V_{CC} conditions for both low-to-high and high-to-low transitions as capacitance increases.



Conditions:
T_A = 70°C
V_{CC} = 4.75V

Figure 12. 85C060 t_{PD} vs C_L

t_{PD} vs Number of P-Terms Programmed

As additional p-terms for a macrocell are used, internal device loading causes propagation delay to increase or decrease slightly. Figure 13 shows this slowdown for the 85C060 at room temperature conditions with 1 output switching.

Provided in this figure is the worst case variation between low-to-high and high-to-low graphs (including variation due to logic inversion (ON or OFF). This provides an envelope inside which any other conditions will fall. Designers can see that the worst case variation is less than 1 ns. Further characterization shows this variation does not change very much over temperature. The absolute values shown in this figure are from "typical" 85C060 devices.

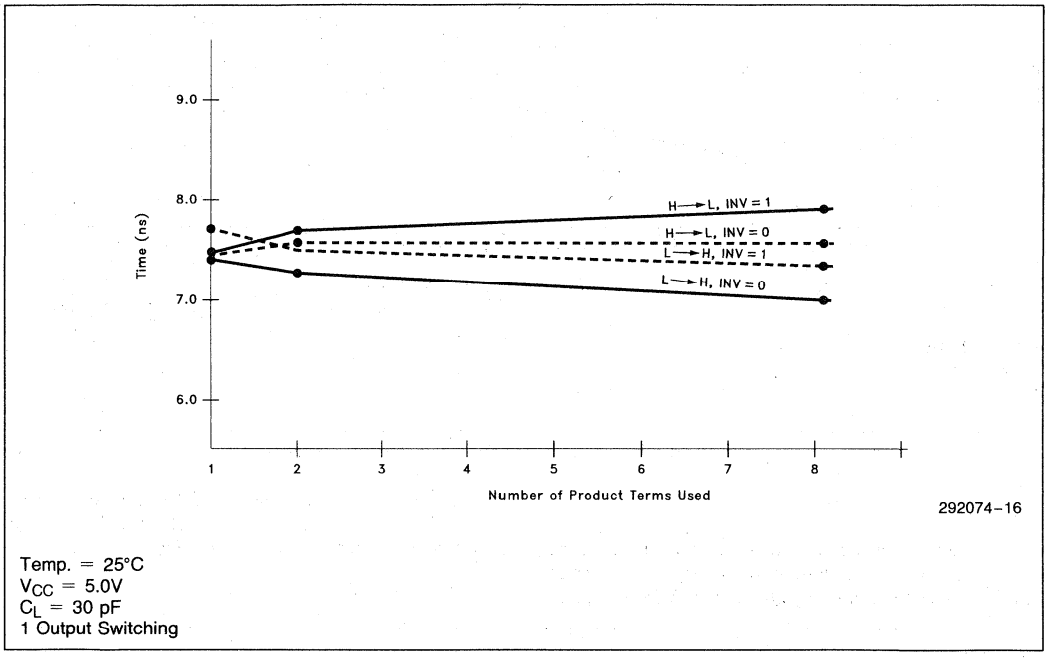


Figure 13. 85C060 t_{PD} vs Number of P-Terms Programmed

Synchronous Register Characteristics

t_{CO_1} (clock to register output valid) is in the 2.0 ns to 7.0 ns range for the 85C060-12. As shown in Figure 14 the t_{CO_1} value varies only slightly (less than 700 ps) over the operating temperature range of this device.

This data demonstrates to designers that the t_{CO_1} value is fairly constant over temperature, which eases their task in performing a worst-case circuit analysis. This figure shows the t_{CO_1} characterization at V_{CC} = 4.75V (the worst-case value). There is virtually no variation in H ⟶ L characterization with V_{CC} varied from 4.75V to 5.25V. There is a small variation (less than 0.5 ns) in the L ⟶ H transition due to V_{CC} changes. The result is a t_{CO_1} value that does not change much over temperature and V_{CC}.

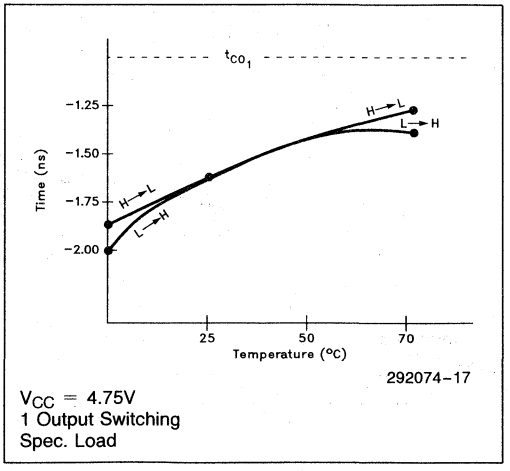


Figure 14. t_{CO_1} vs Temperature

Register-to-Register Skew

When registers within the same programmable logic device are clocked (using the synchronous clock input), ideally all outputs would change state simultaneously. This does not reflect the reality of differences in internal clock routing and ground path differences. Thus, there is some skew between outputs. As long as this skew remains small there is no impact on system design. Large skews can cause a need for additional synchronization logic and re-evaluation of system timing constraints.

Due to its high-speed double metal process, the output skew on the Intel 85C060 μPLD is very tight. Typical skew between fastest and slowest register within one "bank" (i.e., clocked by the same clock pin) is shown in Table 7.

Table 7. 85C060 Register-to-Register Skew Characterization

0°C		70°C	
High to Low (ps)	Low to High (ps)	High to Low (ps)	Low to High (ps)
120	310	120	340

ASYNCHRONOUS REGISTER OPERATION CHARACTERISTICS

In addition to combinational logic and synchronous registered logic, the 85C060 μPLD can implement asynchronously-clocked registered logic. This means a product (AND) term can be used to control the register in any macrocell. Each macrocell has a separate product term which can be used for this purpose. Also, as discussed in Section 3 there are a separate set of A.C. specifications for asynchronous register operation. Two of the key specifications are the asynchronous clock-to-output delay (t_{ACO_1}) and the asynchronous register setup time (t_{ASU}).

t_{ACO_1} Characteristics

Knowing how this specification varies over supply voltage (V_{CC}) and temperature may be useful to a designer concerned with detailed system timing analysis. Figure 15 shows this t_{ACO_1} characterization (H ⟶ L transition only) and how small the changes are over both of these variables. This figure shows t_{CO_1} will increase if V_{CC} is decreased or if system temperature increases, although temperature has a much greater impact on its value. The total variation over both of these parameters is fairly small (less than 2 ns) showing the solid design of the asynchronous circuitry of the 85C060.

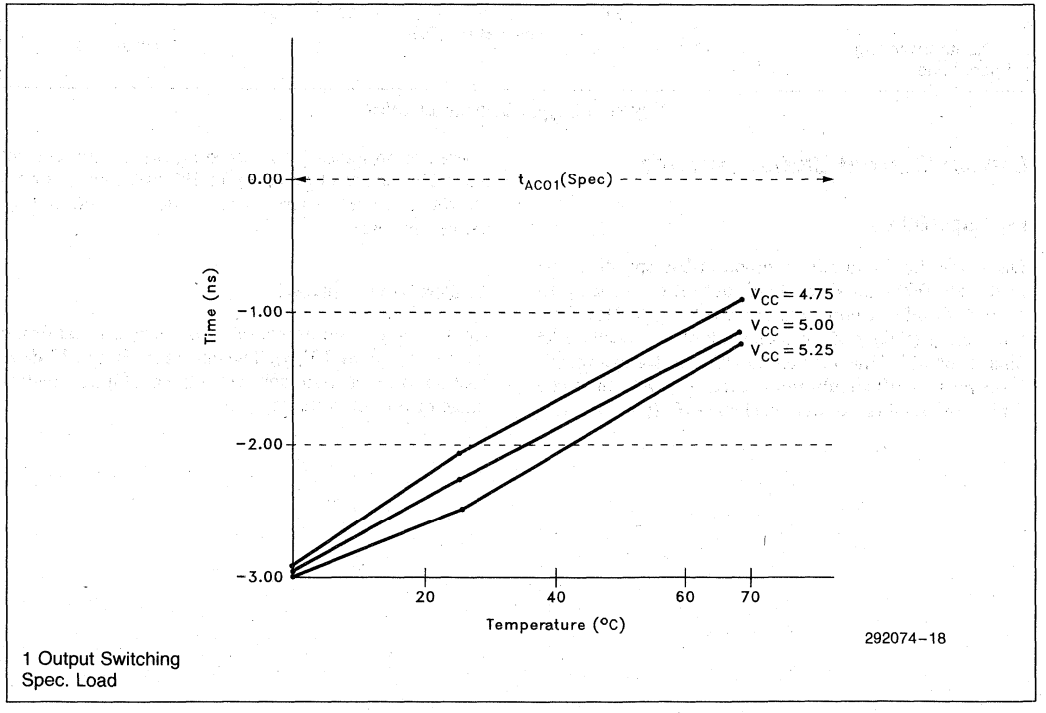


Figure 15. 85C060 t_{ACO_1} Characteristics (H ⟶ L)

t_{ASU} Characteristics

Another important specification related to asynchronous register operation is t_{ASU}, the register setup time. A designer performing circuit analysis may need to know how this may vary over temperature and supply voltage. Figure 16 presents the characterization of this specification. Of note are the very small changes over both temperature (less than 50 ps over 0°C–70°C range) and supply voltage (less than 350 ps variation over a 4.75V–5.25V range). Both of these show the exceptional stability of this circuitry.

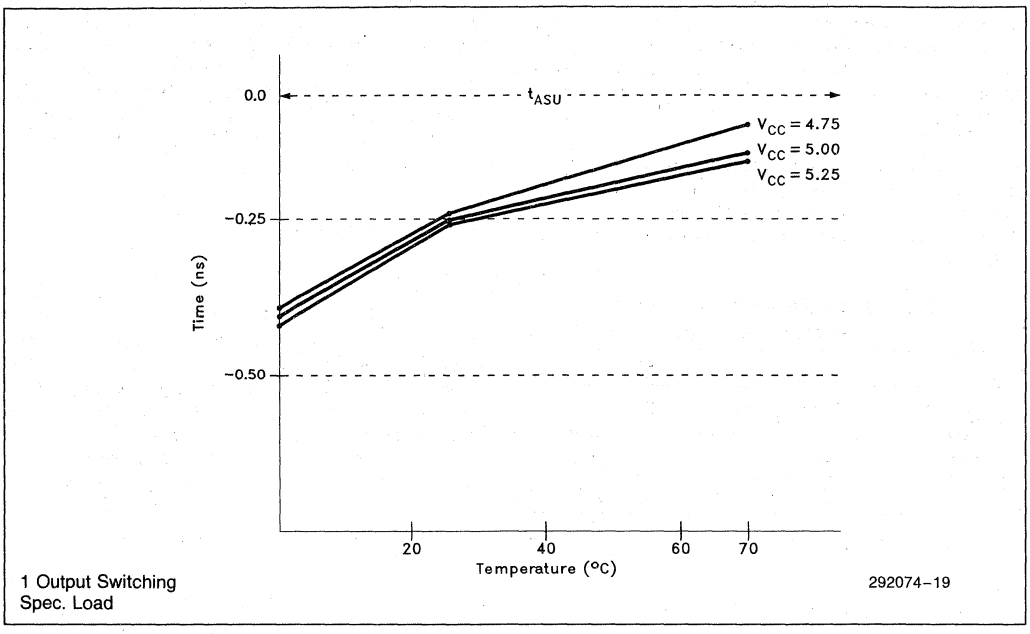


Figure 16. t_{ASU} Characteristics

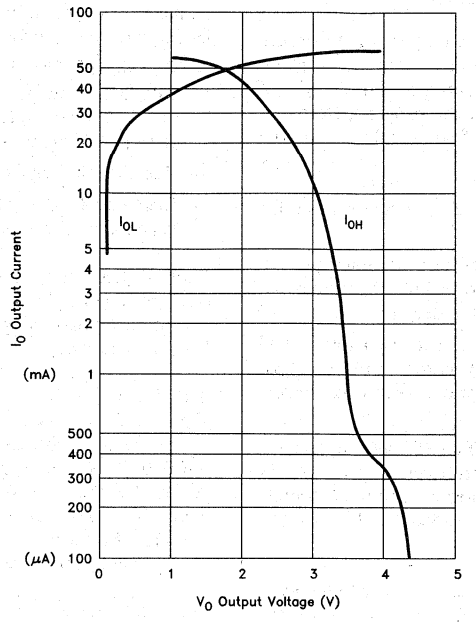
Output Current Characteristics

I_{OL} Capabilities

Designers should note the method of I_{OL} specifications in the 85C060 data sheet. Each output can sink up to 12 mA (while meeting the specified V_{OL}). However, the average (DC) load for each bank of macrocells (I/O pins) is 64 mA. This means the DC load for macrocells 1–8 (pins 3–10) should not exceed 64 mA. This same limitation applies to macrocells 9–16 (pins 15–22). There are no issues with exceeding the 64 mA current for short periods of time, (up to 192 mA total), however, the average current consumption should not be above this level.

Output Drive Current

Another important aspect of output driver capability is the I_O vs V_{OL} and V_{OH}. The curves in Figure 17 demonstrate the current source/sink capabilities over a range of output voltages.



Conditions:
$T_A = +80°C$
$V_{CC} = 4.75V$

Figure 17. 85C060 Output Drive Current

Other Design Considerations

High-performance CMOS devices such as the 85C060 are capable of driving large loads at fast edge rates. This capability means that noise control must be an important consideration during system design. Multilayer PC boards utilizing ground and power planes provide low resistance and low inductance connections between the power sources and devices.

System noise control also requires good decoupling capacitance. Boards with power and ground planes but no decoupling capacitors can still have noise problems. High speed transients of devices may demand up to 500 mA of current, which can result in a volt or more of switching noise on the local supply lines. Decoupling capacitors can help prevent this performance degradation by providing a local power source during output transitions. With the addition of decoupling capacitors, it is possible to reduce the local supply noise to 200 mV or less.

Capacitor selection is important for this application since the frequencies involved in high-speed systems can exceed 100 MHz. High-frequency capacitors are called for. The capacitors should provide low series inductance; leadless chip caps are the best choice, with short leaded capacitors as a more available second choice. The equivalent circuit for a capacitor is a series resonant circuit. If the inductive element in the capacitor is too high, the capacitor will appear inductive at high frequencies.

Assuming that everything possible has been done to manage noise on the supply lines outside the device, internal noise can still be a problem. The internal noise generated during switching transients is caused by output buffer design, package design, and output loading. Some suggestions for reducing noise are as follows:

- Select a low-inductance package such as PLCC.
- Reduce the output loading.
- Reduce the number of simultaneously switching outputs.
- Limit the voltage swing to 0V–3V by correctly terminating outputs with resistors to ground.

Designing high-speed P.C. boards requires closer attention to design issues that are not as important for slower systems. These elements include:

- Termination of transmission lines.
- Clock signal routing.
- Power distribution and heat dissipation.

5.0 APPLICATION IDEAS

The 85C060's high speed and architectural flexibility can be applied to many of today's system design problems. Applications requiring storage and decode of more than 8 bits, high-performance registers, state machines, and control logic all find a solution in the 85C060 μPLD.

The following is a list of several application ideas for the 85C060:

(a) Bus Controller: For 386, 486 or other microprocessors. The PLD can integrate bus state tracking, READY logic and bus status decoding.

(b) Shared Memory Arbitration/Bus Control: Useful for intelligent EISA/MCA cards, multiprocessor system designs and applications requiring sharing of resources on a common bus.

(c) Custom Control Register: The 85C060 implements an 8-bit, high-speed, bidirectional register. Additional decode of each register bit provides discrete control signals to microprocessor and peripheral devices.

These applications will be discussed further to help you fully use the architecture and performance of the 85C060.

5.1 80386DX Bus Controller

In every 80386DX microprocessor system, bus control logic must be implemented to provide an interface to system peripherals, I/O devices, and system memory. The bus controller requirements can vary in complexity depending on system performance, memory hierarchy, and other factors. Figure 18 shows an example of an 80386 subsystem with an 85C060 bus controller. The bus controller decodes processor status signals, provides system peripheral/I/O device control signals, and manages bus cycle timing. In this case the bus controller implemented performs the following functions:

— Decode of 386 bus status signals to generate EPROM control, Interrupt Acknowledge cycles and I/O Read/Write signals. In addition, the I/O READY indication is generated.

— Bus Transceiver Control (OE Control)

— Bus State Tracking functions to determine if the bus is active or if a pipelined bus cycle is occurring.

Each of these three functions represent interdependent state machines. These state machines were designed and implemented using the ABEL logic compiler from Data I/O Corp.

The design file for the 386 bus controller is shown in Figure 19. Note the device name used "E0600" is valid for all devices with architectures compatible with the 85C060. This would include the Intel 5C060 and the Altera EP600, 610, 630, among others. This ABEL source file was implemented in the same fashion as common PALs or GALs, although the extended feature of the 85C060 are available to the designer.

One note, when using test vectors with an ABEL source file be sure to toggle the proper clock or clocks. The 85C060 has two synchronous clock inputs and when in doubt, toggle both. Another ABEL feature which is relevant to the 85C060 is the "ISTYPE" statement. This can be used by the designer if specific register types, feedback or invert options are required for any 85C060 implementation. Upon successful compilation of this source file by ABEL, a .DOC file (see Figure 20) was generated (along with the required JEDEC file) which shows the reduced equations and device pinout. This bus state tracker will work for 80386DX designs up to 33 MHz (CLK2 = 66 MHz) due to the 85C060s state machine frequency specification (F_{CNT_1}) of 66 MHz.

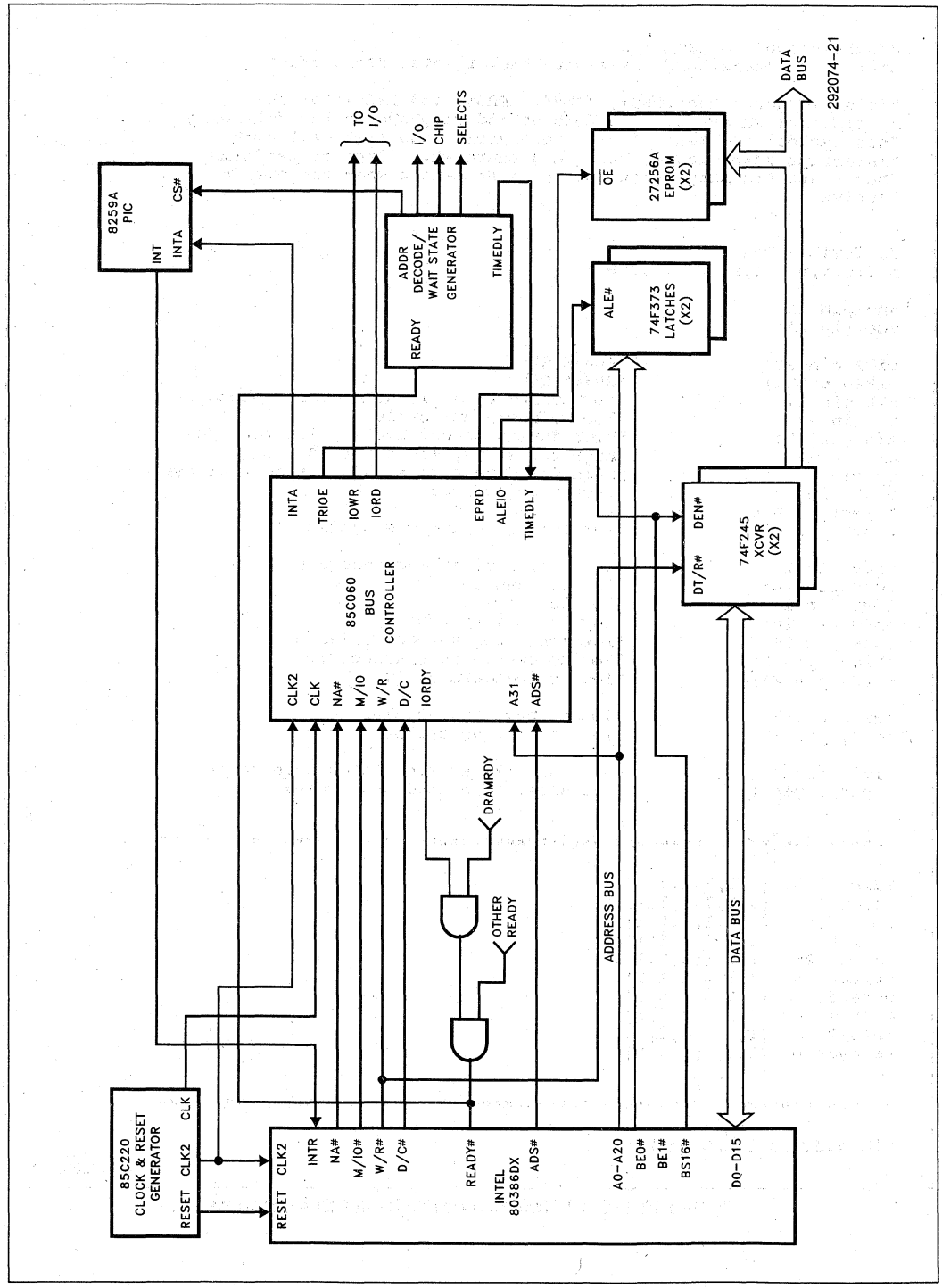


Figure 18. Typical 80386DX Microprocessor Subsystem

```
module pst060;  flag '-r3';
title 'io controller/ bus state tracker intel corporation'

"This pld generates IORD#, IOWR#, EPRD#, and INTA# for the
"peripheral subsystem.  It decodes and responds to the following
"bus cycles: i/o read, i/o write, memory read (with A31 high),
"interrupt acknowledge, halt, and shutdown.  Also, it performs
"bus state tracking functions which determine when the bus is
"active.


U1 device 'E0600';
h,l,c,x,p = 1,0,.C.,.X.,.P.;

gnd pin 12;
vcc pin 24;

clk2 pin 13;            "80386 CLK2
clk2a pin 1;            "80386 CLK2
clk pin 2;              "low during phase 1, high during phase 2
na pin 3;               "low to begin bus cycles
mio pin 23;             "high during memory cycles, low for i/o
wr pin 5;               "high for write, low for read
dc pin 6;               "high for data cycles, low for control cycles
pa31 pin 7;             "processor address A31
timedly pin 8;          "time delay input
buscyc pin 9;           "low during active bus cycles

recv pin 15;            "low during float and recovery
iord pin 16;            "low to read io
iowr pin 17;            "low to write io
eprd pin 18;            "low to read eproms
inta pin 19;            "low for interrupt acknowledge
trioen pin 20;          "low to enable io transceiver
iordy pin 21;           "low to indicate ready

ads pin 14;             "low to begin bus cycles
ready pin 11;           "low to end bus cycles

aleio pin 22;           "high to make address latch transparent
pipecyc pin 4;          "low after pipelined bus cycles


"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""

idle    =  [1,1,1,1,1,1];
ioread1 =  [0,1,1,1,1,1];
ioread2 =  [0,1,1,1,0,1];
iowrite1=  [1,0,1,1,1,1];
iowrite2=  [1,1,1,1,0,1];
epread1 =  [1,1,0,1,1,1];
epread2 =  [1,1,0,1,0,1];
intak1  =  [1,1,1,0,1,1];
intak2  =  [1,1,1,0,0,1];
recover =  [1,1,1,1,1,0];


"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""

"io address latch enable
```

292074-32

Figure 19. 85C060 ABEL Source File for 386 Bus Controller

```
equations  !aleio :=    (!iord & clk) #
                        (!iowr & clk) #
                        (!inta & clk) #
                        (!aleio & !clk);

"io transceiver enable

state_diagram [trioen];
state 1:        "idle
        if (na & !buscyc & !mio & !pa31 & recv & clk) then 0
        else if (na & !buscyc & mio & pa31 & recv & clk) then 0
        else 1;

state 0:        "enable transceiver between processor and peripherals

        if (!iordy & clk) then 1
        else if (buscyc & clk) then 1
        else 0;

"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""

"io state machine

state_diagram [iord, iowr, eprd, inta, iordy, recv];
state idle:
  case  na & !buscyc & pa31 & !wr & clk: epread1;
        na & !buscyc & !pa31 & !mio & dc & wr & clk: iowrite1;
        na & !buscyc & !pa31 & !mio & dc & !wr & clk:ioread1;
        na & !buscyc & !pa31 & !mio & !dc & !wr & clk: intak1;
        na & !buscyc & mio & !dc & wr & clk: iowrite2;  "halt
  endcase;

state epread1:  if (!timedly & clk) then epread2 else epread1;
state epread2:  if (clk) then idle else epread2;
state iowrite1: if (!timedly & clk) then iowrite2 else iowrite1;

state iowrite2: if (!mio & clk) then recover
                else if (mio & clk) then idle
                else iowrite2;
state ioread1:  if (!timedly & clk) then ioread2 else ioread1;
state ioread2:  if (clk) then recover else ioread2;
state intak1:   if (!timedly & clk) then intak2 else intak1;
state intak2:   if (clk) then recover else intak2;
state recover:  if (!timedly & clk) then idle else recover;

"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""

"bus cycle tracking

state_diagram [buscyc, pipecyc]
state[1,1]:     "idle
        if (!ads & clk) then [0,1]
        else [1,1];

state [0,1]:    "active
        if (!ready & ads & clk) then [1,1]
        else if (!ready & !ads & clk) then [1,0]
        else [0,1];
```

292074-33

Figure 19. 85C060 ABEL Source File for 386 Bus Controller (Continued)

```
state [1,0]:    "pipelined
       if(clk) then [0,1]
       else [1,0];

state [0,0]:    "illegal
       goto [1,1];


"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
test_vectors ([clk2,clk2a,clk,na,mio,wr,dc,pa31,timedly,buscyc]  ->
              [iord,iowr,eprd,inta,iordy,recv]);

[c,c,h,h,h,h,h,h,h,h]  ->  [h,h,h,h,h,h];       "idle
[c,c,h,h,h,h,h,h,h,h]  ->  [h,h,h,h,h,h];       "idle
[c,c,h,h,h,h,h,h,h,h]  ->  [h,h,h,h,h,h];       "idle

[c,c,h,x,x,x,x,x,x,l]  ->  [h,h,h,h,h,h];       "preload buscyc
[c,c,h,h,h,l,l,h,h,l]  ->  [h,h,l,h,h,h];       "eprom read
[c,c,h,h,h,l,l,h,h,l]  ->  [h,h,l,h,h,h];       "eprom read
[c,c,h,h,h,l,l,h,h,l]  ->  [h,h,l,h,h,h];       "eprom read
[c,c,h,h,h,l,l,h,h,l]  ->  [h,h,l,h,h,h];       "eprom read
[c,c,h,h,h,l,l,h,l,l]  ->  [h,h,l,h,l,h];       "eprom read
[c,c,h,h,h,h,h,h,h,h]  ->  [h,h,h,h,h,h];       "idle

[c,c,h,h,l,l,h,l,h,l]  ->  [l,h,h,h,h,h];       "io read
[c,c,h,h,l,l,h,l,h,l]  ->  [l,h,h,h,h,h];       "io read
[c,c,h,h,l,l,h,l,h,l]  ->  [l,h,h,h,h,h];       "io read
[c,c,h,h,l,l,h,l,h,l]  ->  [l,h,h,h,h,h];       "io read
[c,c,h,h,l,l,h,l,l,l]  ->  [l,h,h,h,l,h];       "io read
[c,c,h,h,h,h,h,h,h,h]  ->  [h,h,h,h,h,l];       "recovery
[c,c,h,h,h,h,h,h,h,h]  ->  [h,h,h,h,h,l];       "recovery
[c,c,h,h,h,h,h,h,l,h]  ->  [h,h,h,h,h,h];       "idle

[c,c,h,h,l,h,h,l,h,l]  ->  [h,l,h,h,h,h];       "io write
[c,c,h,h,l,h,h,l,h,l]  ->  [h,l,h,h,h,h];       "io write
[c,c,h,h,l,h,h,l,h,l]  ->  [h,l,h,h,h,h];       "io write
[c,c,h,h,l,h,h,l,h,l]  ->  [h,l,h,h,h,h];       "io write
[c,c,h,h,l,h,h,l,l,l]  ->  [h,h,h,h,l,h];       "io write
[c,c,h,h,l,h,h,h,h,h]  ->  [h,h,h,h,h,l];       "recovery
[c,c,h,h,h,h,h,h,h,h]  ->  [h,h,h,h,h,l];       "recovery
[c,c,h,h,h,h,h,h,l,h]  ->  [h,h,h,h,h,h];       "idle

[c,c,h,h,l,l,l,l,h,l]  ->  [h,h,h,l,h,h];       "interrupt ack
[c,c,h,h,l,l,l,l,h,l]  ->  [h,h,h,l,h,h];       "interrupt ack
[c,c,h,h,l,l,l,l,h,l]  ->  [h,h,h,l,h,h];       "interrupt ack
[c,c,h,h,l,l,l,l,h,l]  ->  [h,h,h,l,h,h];       "interrupt ack
[c,c,h,h,l,l,l,l,l,l]  ->  [h,h,h,l,l,h];       "interrupt ack
[c,c,h,h,h,h,h,h,h,h]  ->  [h,h,h,h,h,l];       "recovery
[c,c,h,h,h,h,h,h,h,h]  ->  [h,h,h,h,h,l];       "recovery
[c,c,h,h,h,h,h,h,l,h]  ->  [h,h,h,h,h,h];       "idle

[c,c,h,h,h,h,l,l,h,l]  ->  [h,h,h,l,h,l];       "halt or shutdown

[c,c,h,h,h,h,h,h,h,h]  ->  [h,h,h,h,h,h];       "idle
[c,c,h,h,h,h,h,h,h,h]  ->  [h,h,h,h,h,h];       "idle

"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""

test_vectors ([clk2a, clk, ads, ready]  ->  [buscyc, pipecyc])
```

292074-34

Figure 19. 85C060 ABEL Source File for 386 Bus Controller (Continued)

```
        [p,h,l,l]  ->  [x,x];
        [c,l,h,h]  ->  [x,h];          "idle-busy-idle
        [c,h,h,l]  ->  [h,h];
        [c,l,h,h]  ->  [h,h];
        [c,h,l,h]  ->  [l,h];
        [c,l,h,l]  ->  [l,h];
        [c,h,h,l]  ->  [h,h];

        [c,l,l,h]  ->  [h,h];          "idle-busy-pipe-busy
        [c,h,l,h]  ->  [l,h];
        [c,l,h,l]  ->  [l,h];
        [c,h,l,l]  ->  [h,l];
        [c,l,l,h]  ->  [h,l];
        [c,h,h,h]  ->  [l,h];

        [c,h,h,l]  ->  [h,h];          "idle-busy-busy-idle
        [c,l,h,h]  ->  [h,h];
        [c,h,l,h]  ->  [l,h];
        [c,l,l,h]  ->  [l,h];
        [c,h,h,h]  ->  [l,h];
        [c,l,h,h]  ->  [l,h];
        [c,h,h,h]  ->  [l,h];
        [c,l,h,c]  ->  [l,h];
        [c,h,h,l]  ->  [h,h];
        [c,l,h,h]  ->  [h,h];

end pst060;
```

292074-35

Figure 19. 85C060 ABEL Source File for 386 Bus Controller (Continued)

```
                                                                    Page 1
ABEL(tm) 3.20A  -  Document Generator              17-Jul-90 06:26 PM
io controller/ bus state tracker intel corporation
Equations for Module pst060

Device U1


- Reduced Equations for device U1:

    !aleio := (!clk & !aleio # clk & !inta # clk & !iowr # clk & !iord);

    !trioen := (!clk & !trioen
                # !buscyc & !trioen & iordy
                # clk & na & mio & pa31 & !buscyc & recv & trioen
                # clk & na & !mio & !pa31 & !buscyc & recv & trioen);


    !iord := (!clk & recv & !iord & iowr & eprd & inta
              # recv & !iord & iowr & eprd & inta & iordy
              # clk & na & !mio & !wr & dc & !pa31 & !buscyc & recv & iowr &
            eprd & inta & iordy);

    !iowr := (!clk & recv & iord & !iowr & eprd & inta & iordy
              # timedly & recv & iord & !iowr & eprd & inta & iordy
              # clk & na & !mio & wr & dc & !pa31 & !buscyc & recv & iord &
            iowr & eprd & inta & iordy);

    !eprd := (!clk & recv & iord & iowr & !eprd & inta
              # recv & iord & iowr & !eprd & inta & iordy
              # clk & na & !wr & pa31 & !buscyc & recv & iord & iowr & inta
            & iordy);

    !inta := (!clk & recv & iord & iowr & eprd & !inta
              # recv & iord & iowr & eprd & !inta & iordy
              # clk & na & !mio & !wr & !dc & !pa31 & !buscyc & recv & iord
            & iowr & eprd & iordy);

    !iordy := (!clk & recv & iord & iowr & eprd & !iordy
               # clk & !timedly & recv & iord & iowr & eprd & !inta & iordy

               # !clk & recv & iowr & eprd & inta & !iordy
               # clk & !timedly & recv & !iord & iowr & eprd & inta & iordy

               # clk & !timedly & recv & iord & !iowr & eprd & inta & iordy

               # !clk & recv & iord & iowr & inta & !iordy
               # clk & !timedly & recv & iord & iowr & !eprd & inta & iordy

               # clk & na & mio & wr & !dc & !buscyc & recv & iord & iowr &
             eprd & inta & iordy);

    !recv := (!clk & !recv & iord & iowr & eprd & inta & iordy
              # timedly & !recv & iord & iowr & eprd & inta & iordy
              # clk & recv & iord & iowr & eprd & !inta & !iordy
              # clk & recv & !iord & iowr & eprd & inta & !iordy
              # clk & !mio & recv & iowr & eprd & inta & !iordy);
```

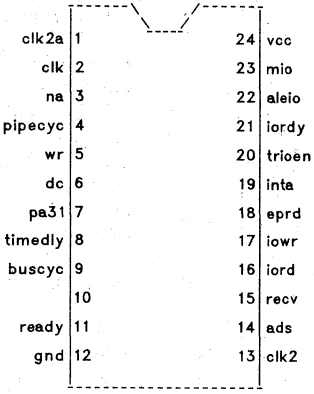
292074-36

Figure 20. 386-PST .DOC File

```
Device U1

    !buscyc := (clk & buscyc & !pipecyc
               # !clk & !buscyc & pipecyc
               # !buscyc & ready & pipecyc
               # clk & buscyc & !ads);

    !pipecyc := (!clk & buscyc & !pipecyc
                # clk & !buscyc & !ads & !ready & pipecyc);
```

292074-37

E0600



292074-22

Figure 20. 386-PST .DOC File (Continued)

5.2 Shared Memory Arbitration/Bus Control

Sharing resources (usually memory) on a common bus is becoming more and more common in today's system designs. Not only are many systems, including personal computers, being implemented as multiprocessors, but peripheral controllers are becoming more intelligent and capable of controlling the bus by themselves. Thus, arbitration logic is required to determine which processor/controller currently has control of each shared resource. In addition, the designer may want to incorporate a variety of associated functions including:

- READY Logic (Bus Cycle Control)
- Address Pipelining Support
- Memory Burst Control
- Wait-State Generation
- I/O Chip Select Logic
- Bus Throttling Logic
- DRAM Control/Refresh Logic (if used instead of SRAM)
- EPROM Control Logic (if microprocessor code is not in SRAM)

The example to be discussed here is an intelligent EISA communications add-in card (see Figure 21). The arbitration logic must decide if the communications controller, on-board microprocessor or EISA bus controller has access to the on-board static RAM (SRAM). The on-board SRAM mainly acts as a high-speed data buffer for the communications controller to off-load the EISA bus by providing block size transfers. The on-board microprocessor initializes the EISA control logic and communications controller and provides handling of local interrupts/error conditions. A block diagram of the system and required logic are provided by Figures 21 and 22, respectively. The "communications controller" may be implementing an ISDN, high-speed serial, Ethernet, FDDI or other communications link. Often these devices can themselves control the local bus and provide DMA capabilities to move data to and from memory. Also, many of these controllers have on-board data FIFOs that may necessitate a need for bursting data to/from the local memory.

Each of the three bus control-capable devices is assumed to have "Bus Request" and "Bus Grant" signals that are routed to the arbitration logic. The 85C060 provides 8 product terms (in addition to separate OE and RESET product terms) in each macrocell, which allows for increased flexibility in choosing an arbitration scheme. Common schemes include fixed priority, rotating (or last granted, lowest priority), or First-Come, First-Serve. In this example a rotating priority is implemented to assure a balance of accesses between the devices, and to decrease the worst-case service time to help prevent data underflow/overflow. Figure 23 shows the state diagram of the 3-way bus arbitration logic. This method is easily expandible if more bus masters were present, as would be the case for DRAM Refresh requests and/or multiple communications controllers or in a multiprocessing system.

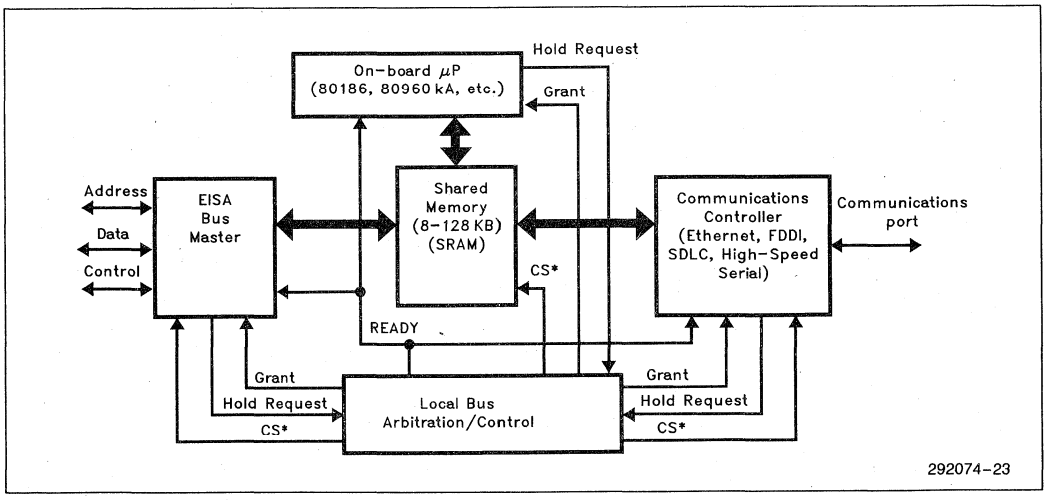


Figure 21. Intelligent EISA Communications Add-In Card

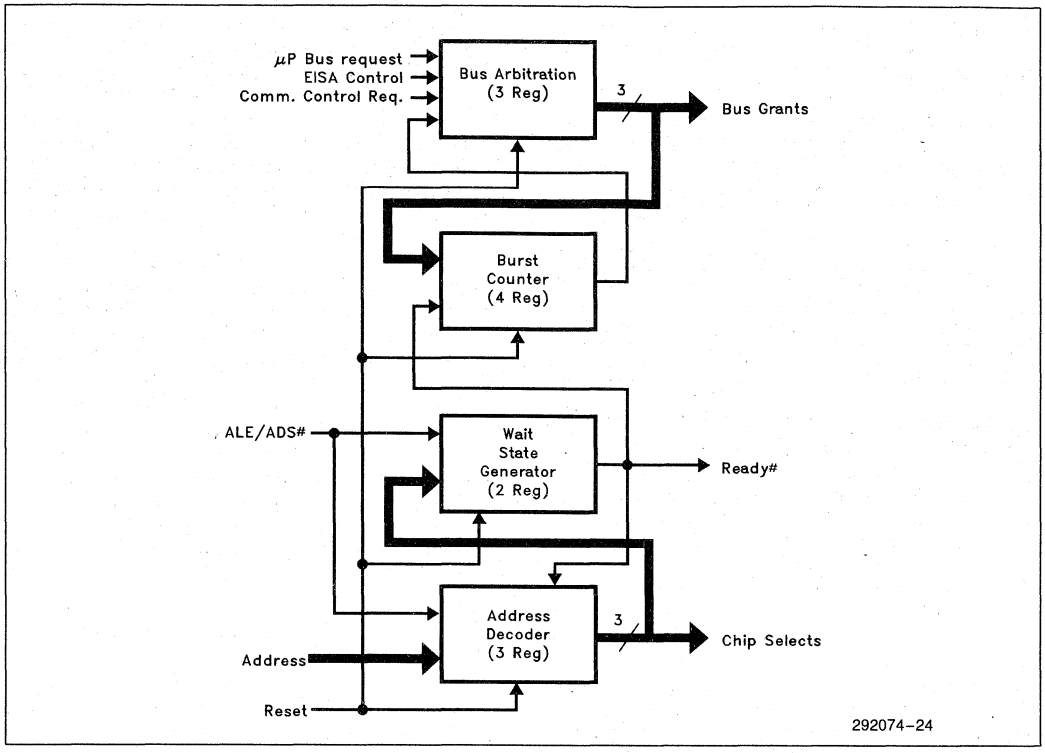


Figure 22. 85C060 Shared Memory Arbitration/Bus Control Logic Implementation

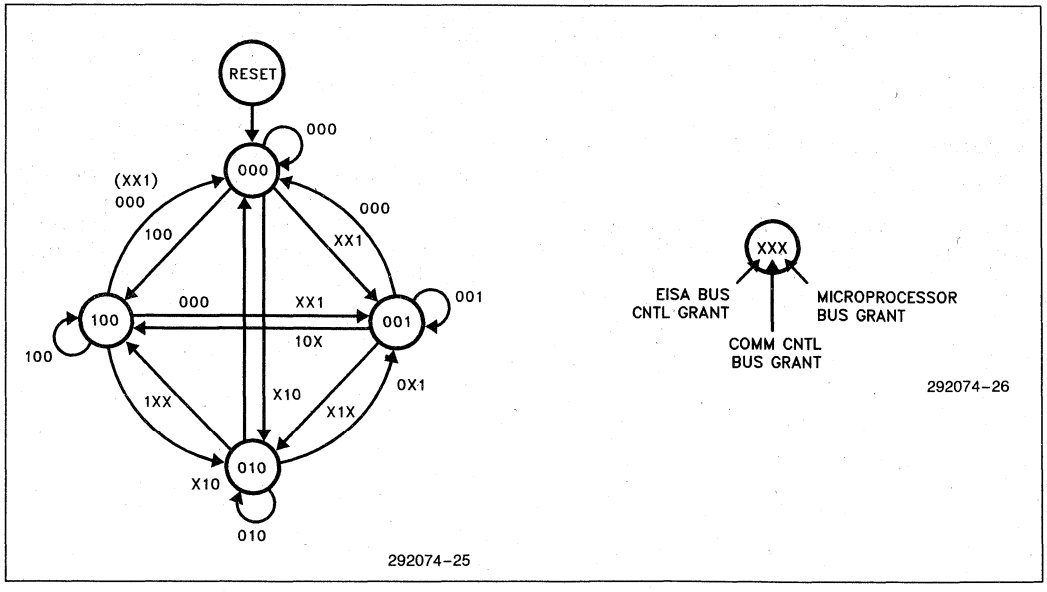


Figure 23. Bus Arbitration State Machine—Rotating Priority

This state machine could easily be changed to default to grant the bus to the microprocessor when no device requests are active.

In this implementation each bus grant allows the granted device to execute a "burst transfer". Thus, a counter is needed to keep track of the following:

Device	Burst Transfer	Data Transfer Size
Microprocessor	1	16-bit
Comm Controller	8	16-bit
EISA Controller	8	16-bit

The burst counter counts the number of times READY ("bus cycle completed") is generated and ends the transaction after the appropriate number of transfers, which signals the arbitration logic to re-arbitrate the local bus. This means the size of each transfer in the burst is transparent to the burst logic. Logic is also required to generate local chip selects for the RAM, Comm Controller and EISA Bus Controller. The memory map for each device is the same (i.e., the SRAM, etc. are at the same addresses for each device) and there are very few addresses required to implement this system, therefore the chip selects are generated from two high-order address lines. These chip selects are routed to the appropriate devices and used internally by the wait-state generator, which counts the number of cycles programmed for each chip select before activating the READY signal. This implementation shows relatively few inputs are needed, but many registers (both for internal and external signals) are required as summarized below:

Inputs = 7
Registers (Internal) = 6
Outputs = 7
} Total I/O's = 20

An added feature of the 85C060 architecture is the use of two separate clock pins. This means half of the macrocells have CLK1 (pin 1) as the source for their synchronous clock and the other half have CLK2 (pin 13) as their source for the synchronous clock. The advantage of this "dual" clocking can be very useful in this design. The system clock may be run at 25 MHz and therefore this clock would be used for the Grant and Chip Select logic. It may be desireable to have a higher clock rate, such as 50 MHz, for the wait state generator to have a finer time granularity for wait times. This could increase the bus utilization greatly depending on how closely the actual access times can be matched.

Another feature of the 85C060 architecture that proves useful for this application is the product term in each macrocell for asynchronous clocking. This allows the burst counter to count "READY's" until it reaches the terminal count. Figure 24 provides a Data I/O ABEL source file for the 85C060 in this application. Note this PLD is as easy to design with as PALs or other devices using standard tools such as ABEL.

If more features are required, this existing 85C060 design can be enhanced by a simple upgrade to the Intel 85C090 μPLD. Using this existing design as a baseline, the following requirements are added:

- Pipelined Address Support
- EPROM Select
- Support of "Shadow RAM" for EPROM
- Increase Burst Support to 64 Transfers
- Bus Transceiver Control Logic (to support separate/simultaneous transfers)

Pipelined address support is accomplished by generating an NA (Next Address) signal one cycle before each READY. The EPROM chip select can be generated from the internal chip select logic. Also, a "Shadow RAM" could be supported if the on-board microprocessor had a need to decrease interrupt service times or other functions. At boot-up, the EPROM, containing the microprocessor's instruction code, can be copied into much faster SRAM. Thus, the EPROM would only be accessed after each RESET, and code would be executed out of the faster SRAM. Increasing burst support to 64 transfers would be accomplished by expanding the burst counter from 4 to 6 macrocells.

```
module arb_cntl          flag '-UL', '-r3'
title  'Arbitration/Bus control logic
John Casey      Intel Corp.      July 1990'

"THIS IS A SAMPLE FILE ONLY.  THIS CIRCUIT PROVIDES A GENERAL SOLUTION !!!


arblogic         device   'E0600';


"inputs
clk1             pin 1;          "33 MHz clock input
clk2             pin 13;         "33 MHz clock input
procreq          pin 2;          "microprocessor local bus request
commreq          pin 23;         "comm controller local bus request
eisareq          pin 11;         "on-board eisa controller local bus req.
addr1            pin 14;         "address input - used for chip select dec.

addr2            pin 3;          "address input - used for chip select dec.
ale_             pin 5;          "ale (or ads/) - represents valid addr.
reset            pin 22;         "board reset - used to set initial state

"outputs
procgrant        pin 6;          "processor local bus grant
commgrant        pin 7;          "communications controller local bus grant
eisagrant        pin 8;          "on-board eisa controller local bus grant
ready_           pin 9;          "indicates end of current bus cycle
commcs           pin 10;         "comm controller chip select
eisacs           pin 15;         "local eisa bus controller chip select
sramcs           pin 21;         "sram chip select

"buried functions
burst0           pin 16;         "part of burst counter logic
burst1           pin 17;         "part of burst counter logic
burst2           pin 18;         "part of burst counter logic
burstdone        pin 19;         "active at end of current burst count
wait0            pin 20;         "lower bit of wait state count logic,
                                 "ready represents output of this logic
wait1            pin 4;          "upper bit of wait state count logic

"macrocell control
burst0, burst1, burst2, burstdone       istype 'feed_reg';
wait0, wait1                            istype 'feed_reg';


"busarb valid states (for the bus arbitration state machine)
s0 = ^b000;              "no grant active
s1 = ^b001;              "microprocessor grant active
s2 = ^b010;              "comm controller grant active
s3 = ^b100;              "eisa bus controller grant active

"wait state generator valid states
ws0 = ^b000;    ws1 = ^b010;    ws2 = ^b100;    ws3 = ^b111;

"burst state machine
bs0 = ^b000;    bs4 = ^b100;
bs1 = ^b001;    bs5 = ^b101;
bs2 = ^b010;    bs6 = ^b110;
bs3 = ^b011;    bs7 = ^b111;
```

292074-38

Figure 24. Sample 85C060 ABEL Source File

```
equations

commcs := !addr1 & !addr2;                          "address mapping will vary
commcs.clk = !ale_;                                 "ale is async clock

eisacs := !addr1 & addr2;
eisacs.clk = !ale_;

sramcs := addr1 & !addr2;
sramcs.clk = !ale_;

burstdone = burst0 & burst1 & burst2;               "terminal count completed

burst0.clk = ready_;                        "burst counter async clocked w/ ready
burst1.clk = ready_;
burst2.clk = ready_;

"Reset signal sets all registers low
procgrant.re = reset;   commgrant.re = reset;   eisagrant.re = reset;
commcs.re = reset;      eisacs.re = reset;      sramcs.re = reset;
burst0.re = reset;      burst1.re = reset;      burst2.re = reset;
wait0.re = reset;       wait1.re = reset;


 "busarb state machine uses a rotating (last grant, lowest priority) scheme
 "which allows bus accesses to be balanced.  If no request then returns to
 "state 000 which allows microprocessor to be #1 priority (this is due to
 "requirement to quickly service interrupts.

state_diagram [procgrant, commgrant, eisagrant]
State s0:  if (burstdone & procreq) then s1 else s0;
           if (burstdone & commreq & !procreq) then s2 else s0;
           if (burstdone & eisareq & !commreq & !procreq) then s3 else s0;

State s1:  if (burstdone & commreq) then s2 else s0;
           if (burstdone & eisareq & !commreq) then s3 else s0;
           if (burstdone & procreq & !eisareq & !commreq) then s1 else s0;

State s2:  if (burstdone & eisareq) then s3 else s0;
           if (burstdone & procreq & !eisareq) then s1 else s0;
           if (burstdone & commreq & !procreq & !eisareq) then s3 else s0;

State s3:  if (burstdone & procreq) then s1 else s0;
           if (burstdone & commreq & !procreq) then s2 else s0;
           if (burstdone & eisareq & !commreq & !procreq) then s3 else s0;


"This wait state generator can support accesses up to 4 clocks long.
"The EISA controller and the comm controller each take 4 cycles (2 wait
"states) and the SRAM takes two cycles (zero wait states).

state_diagram [wait0, wait1, ready_]

State ws0:      if (sramcs) then ws3 else ws0;
                if (commcs # eisacs) then ws1 else ws0;

State ws1:      goto ws2;
```

292074-39

Figure 24. Sample 85C060 ABEL Source File (Continued)

```
State ws2:      goto ws3;

State ws3:      goto ws0;


"the burst logic state machine has a programmed burst length for each bus
"grant.  These are 8 transfers for the comm controller (fifo) and EISA bus
"controller and one transfer for the microprocessor.  When terminal count
"is reached BURSTDONE will go active, indicating to the bus arbitration logic
"that it should re-arbitrate the bus.  At each state the bus grant is checked
"to make sure it is still active (i.e. that a full FIFO transfer is necessary).

state_diagram [burst0, burst1, burst2]
state bs0:  if (procgrant) then bs7 else bs0;
           if (commgrant # eisagrant) then bs1 else bs0;

state bs1: if (!commgrant & !eisagrant) then bs7 else bs2;

state bs2: if (!commgrant & !eisagrant) then bs7 else bs3;

state bs3: if (!commgrant & !eisagrant) then bs7 else bs4;

state bs4: if (!commgrant & !eisagrant) then bs7 else bs5;

state bs5: if (!commgrant & !eisagrant) then bs7 else bs6;

state bs6: goto bs7;

state bs7: goto bs0;



"       test_vectors
"            .
"            .
"            .


end arb_cntl;
```

292074-40

Figure 24. Sample 85C060 ABEL Source File (Continued)

3

5.3 High-Speed Custom Control/ Status Register

In many applications there exists a need for a high-speed dedicated control and/or status register. In multiprocessing systems there may be a need for a system control register; in a communications controller there may be a requirement for a system configuration control register. Figure 25 shows this general application idea. There is a frequent requirement in system add-on card and adapter designs for a Read/Write register that also provides discrete status inputs and control outputs.

The destinations of these discrete signals can include a wide variety of functions that need to be under software control. The functions provided by this register can include enable signals, transceiver/mux control and it can also include status indications such as a communications error. The design, configuration and purpose of each bit of this register is individually selectable, which leads into a wide variety of applications.

The Intel 85C060 μPLD is useful in this application due to these key architectural features:

— 16 macrocells (8 for data bus, 8 for control outputs/ inputs)

— separate product term for RESET (to set known power up state)

— four dedicated inputs (necessary for register control)

— pin/register feedback capabilities and OE control (allows specialized I/O capabilities to be implemented)

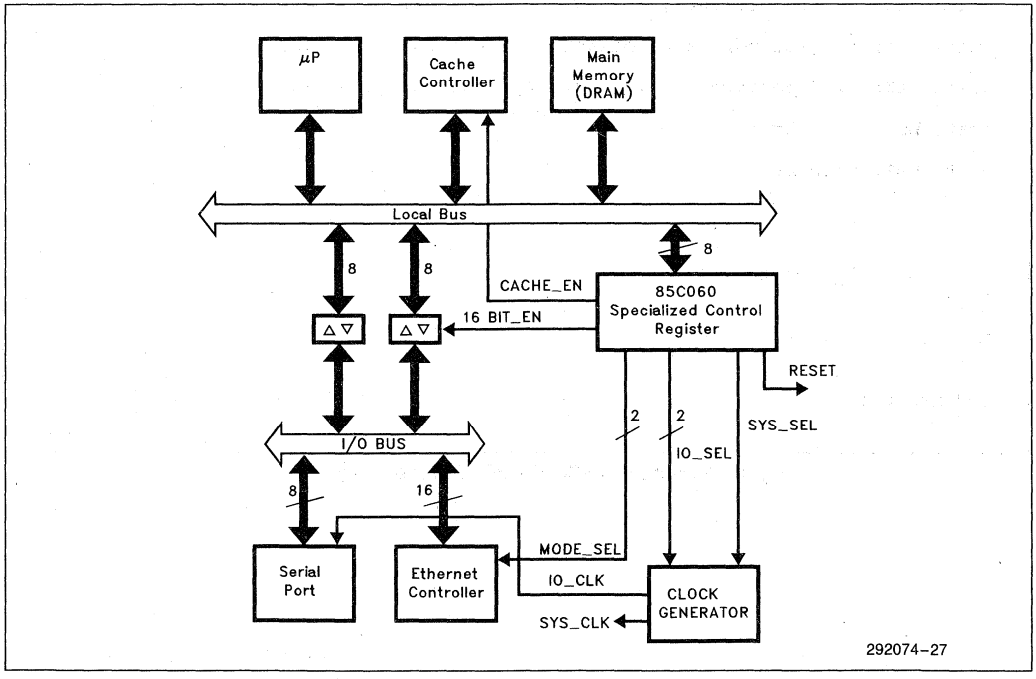


Figure 25. Typical System Requiring Specialized Register for Configuration Control

As shown in Figures 26 and 27, the 85C060 uses these features to implement each bit of this control register. Not only is the architecture of the 85C060 a perfect match for this application, but the performance level is so high (83.3 MHz register speed) that it can meet the zero-wait state requirements for most microprocessors. This provides very high-speed register access via Read-Modify-Write operations.

Some applications may benefit from the power-down mode of the 85C060. If the register implemented is required only periodically, such as at power-on or after system resets, the power-down feature of the 85C060 can be programmed to allow power consumption to be in the 20 μA (typical) range. Decreasing power consumption can have a positive impact on system reliability in addition to decreasing power supply and cooling requirements.

A sample design file is included in Figure 28. This demonstrates the ease of using the advanced architectural capabilities of the 85C060. This design file is written in the Intel ADF (Advanced Design File) format and is compiled by the iPLS II Development Tools (see the PLD Handbook or call the EPLD hotline for more information on Intel development tools). Figure 27 provides details on the actual logic implementation of each of the eight register/control output cells. Each of these cells requires two macrocells of the 85C060. One macrocell forms the register portion and provides the control signal and the other provides the data bus interface (with output enable controlled).

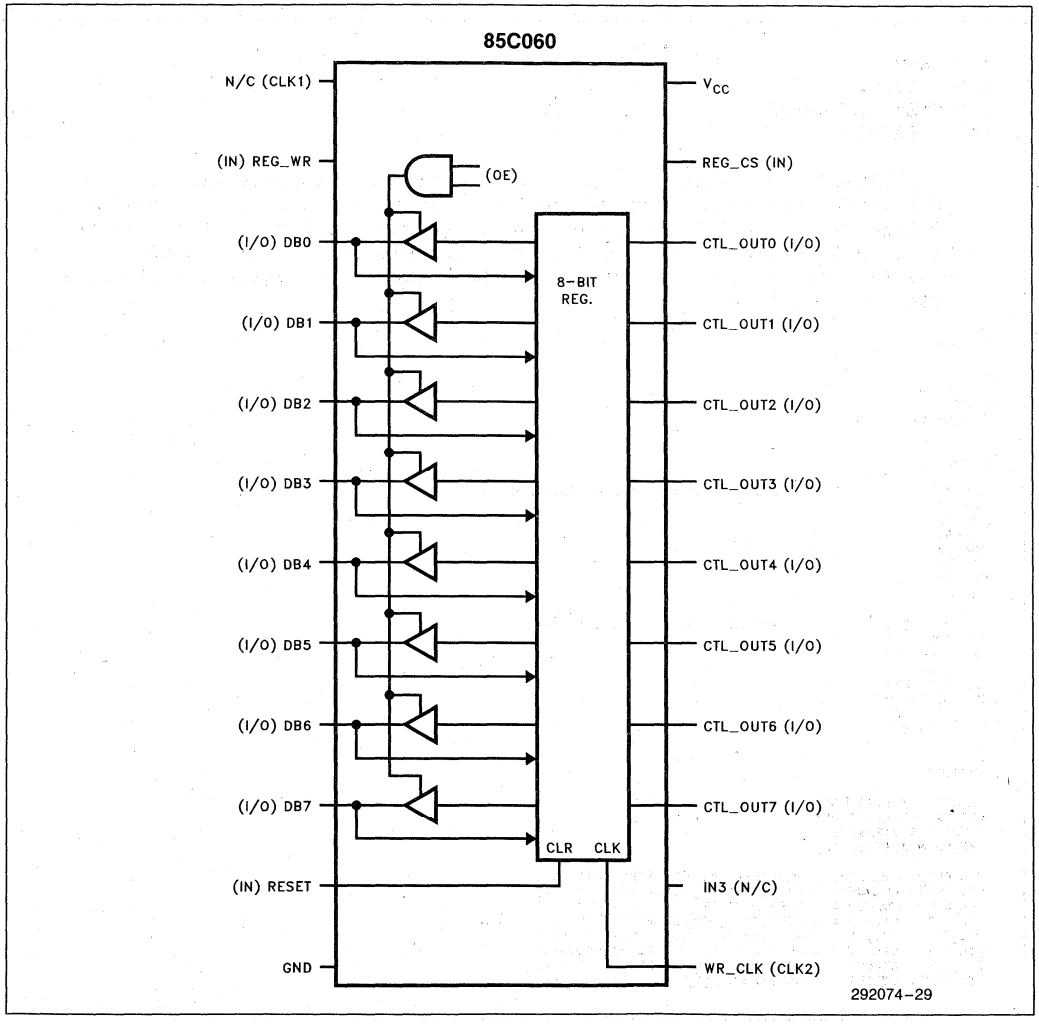


Figure 26. Device Implementation of Control/Status Register

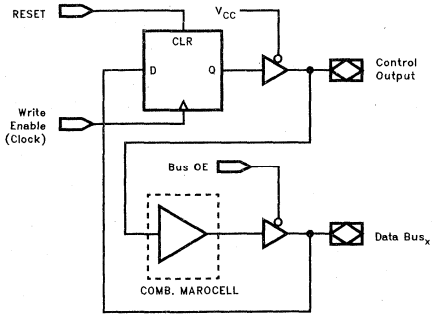


Figure 27. 85C060 Control Register Implementation (1-Bit)

```
J. Casey
Intel Corp.
OPTIONS:  TURBO=ON
PART: 85C060

%  This 85C060 PLD provides a sample design file for implementing a high
   performance customized control register. It includes an 8-bit register
   (D-type flip-flops) and eight discrete control outputs which are routed
   directly to peripherals, bus latches/transceivers, etc. The 8-bit
   register can be read and written by the system processor using the lower
   byte of the data bus.

INPUTS:

WR_CLK @13
REG_WR @2
RESET @11
REG_CS @23


OUTPUTS:

DB0 @3                  % lower byte of system data bus  %
DB1 @4
DB2 @5
DB3 @6
DB4 @7
DB5 @8
DB6 @9
DB7 @10

CTL_OUT7 @15            % Control Register bits 0-7  %
CTL_OUT6 @16
CTL_OUT5 @17
CTL_OUT4 @18
CTL_OUT3 @19
CTL_OUT2 @20
CTL_OUT1 @21
CTL_OUT0 @22


NETWORK:

WR_CLK = INP(WR_CLK)
REG_WR = INP(REG_WR)
RESET = INP(RESET)
REG_CS = INP(REG_CS)

CTL_OUT0, CTLOUT0f = RORF(DB0f,WR_CLK,RESET,GND,VCC)
CTL_OUT1, CTLOUT1f = RORF(DB1f,WR_CLK,RESET,GND,VCC)
CTL_OUT2, CTLOUT2f = RORF(DB2f,WR_CLK,RESET,GND,VCC)
CTL_OUT3, CTLOUT3f = RORF(DB3f,WR_CLK,RESET,GND,VCC)
CTL_OUT4, CTLOUT4f = RORF(DB4f,WR_CLK,RESET,GND,VCC)
CTL_OUT5, CTLOUT5f = RORF(DB5f,WR_CLK,RESET,GND,VCC)
CTL_OUT6, CTLOUT6f = RORF(DB6f,WR_CLK,RESET,GND,VCC)
CTL_OUT7, CTLOUT7f = RORF(DB7f,WR_CLK,RESET,GND,VCC)
```

292074-41

Figure 28. iPLS II Source File for 85C060 Custom Register Design

```
DB0, DB0f = COIF(iDB0,REG_OE)
DB1, DB1f = COIF(iDB1,REG_OE)
DB2, DB2f = COIF(iDB2,REG_OE)
DB3, DB3f = COIF(iDB3,REG_OE)
DB4, DB4f = COIF(iDB4,REG_OE)
DB5, DB5f = COIF(iDB5,REG_OE)
DB6, DB6f = COIF(iDB6,REG_OE)
DB7, DB7f = COIF(iDB7,REG_OE)

EQUATIONS:

iDB0 = CTLOUT0f;
iDB1 = CTLOUT1f;
iDB2 = CTLOUT2f;
iDB3 = CTLOUT3f;
iDB4 = CTLOUT4f;
iDB5 = CTLOUT5f;
iDB6 = CTLOUT6f;
iDB7 = CTLOUT7f;

REG_OE = REG_CS * !REG_WR;

END$
```

292074-42

Figure 28. iPLS II Source File for 85C060 Custom Register Design (Continued)

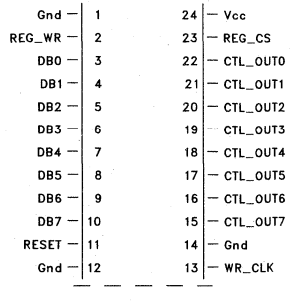
3

A .RPT (Report) file was generated by the Intel iPLS logic compiler. A JEDEC for the 85C060 was also generated by the software. This .RPT file, included in Figure 29, shows the macrocell/p-term usage and device pin-out as the device was implemented.

This application demonstrates the effective combination of architecture and performance provided by the Intel 85C060 μPLD. In essence, the performance increase over existing devices with identical architecture has opened more doors for additional uses. It is now clear how this device can be used in high-speed applications—even when closely coupled to the microprocessor.

```
INTEL Logic Optimizing Compiler Utilization Report                a:CONTROL.rpt
IPLS II FIT Version 2.2 Beta3 Level 4.0i 9/7/88

***** Design implemented successfully

J. Casey
Intel Corp.
OPTIONS:  TURBO=ON
```

85C060



292074-30

Figure 29. iPLS II Report File For The 85C060 Custom Register Design

```
**OUTPUTS**

     Name        Pin        Resource        MCell        PTerms    |   Sync Clock

      DB0          3            COIF           9          1/ 8    |       -
      DB1          4            COIF          10          1/ 8    |       -
      DB2          5            COIF          11          1/ 8    |       -
      DB3          6            COIF          12          1/ 8    |       -
      DB4          7            COIF          13          1/ 8    |       -
      DB5          8            COIF          14          1/ 8    |       -
      DB6          9            COIF          15          1/ 8    |       -
      DB7         10            COIF          16          1/ 8    |       -
 CTL_OUT7         15            RORF           8          1/ 8    |   WR_CLK
 CTL_OUT6         16            RORF           7          1/ 8    |   WR_CLK
 CTL_OUT5         17            RORF           6          1/ 8    |   WR_CLK
 CTL_OUT4         18            RORF           5          1/ 8    |   WR_CLK
 CTL_OUT3         19            RORF           4          1/ 8    |   WR_CLK
 CTL_OUT2         20            RORF           3          1/ 8    |   WR_CLK
 CTL_OUT1         21            RORF           2          1/ 8    |   WR_CLK
 CTL_OUT0         22            RORF           1          1/ 8    |   WR_CLK

**INPUTS**

     Name        Pin        Resource        MCell        PTerms    |   Sync Clock

   WR_CLK         13             CKR           -             -    |       -
   REG_WR          2             INP           -             -    |       -
    RESET         11             INP           -             -    |       -
   REG_CS         23             INP           -             -    |       -

**UNUSED RESOURCES**

     Name        Pin        Resource        MCell        PTerms

        -         14           INPUT           -             -

**PART UTILIZATION**

16/16          Macrocells (100%), 12% of used Pterms Filled
3/ 4           Input Pins (75%)
               PTerms Used 12%
```

Figure 29. iPLS II Report File For The 85C060 Custom Register Design (Continued)

```
Macrocell Interconnection Cross Reference

FEEDBACKS:                    M M M M M M M M   M M M M M M M M
                              0 0 0 0 0 0 0 0   0 1 1 1 1 1 1 1
                              1 2 3 4 5 6 7 8   9 0 1 2 3 4 5 6
CTL_OUT0 . RORF @M1 ->        . . . . . . . .   * . . . . . . .  @22
CTL_OUT1 . RORF @M2 ->        . . . . . . . .   . * . . . . . .  @21
CTL_OUT2 . RORF @M3 ->        . . . . . . . .   . . * . . . . .  @20
CTL_OUT3 . RORF @M4 ->        . . . . . . . .   . . . * . . . .  @19
CTL_OUT4 . RORF @M5 ->        . . . . . . . .   . . . . * . . .  @18
CTL_OUT5 . RORF @M6 ->        . . . . . . . .   . . . . . * . .  @17
CTL_OUT6 . RORF @M7 ->        . . . . . . . .   . . . . . . * .  @16
CTL_OUT7 . RORF @M8 ->        . . . . . . . .   . . . . . . . *  @15

DB0 ..... COIF @M9 -> *       . . . . . . .     . . . . . . . .  @3
DB1 ..... COIF @M10-> .       * . . . . . .     . . . . . . . .  @4
DB2 ..... COIF @M11-> .       . * . . . . .     . . . . . . . .  @5
DB3 ..... COIF @M12-> .       . . * . . . .     . . . . . . . .  @6
DB4 ..... COIF @M13-> .       . . . * . . .     . . . . . . . .  @7
DB5 ..... COIF @M14-> .       . . . . * . .     . . . . . . . .  @8
DB6 ..... COIF @M15-> .       . . . . . * .     . . . . . . . .  @9
DB7 ..... COIF @M16-> .       . . . . . . *     . . . . . . . .  @10


INPUTS:

REG_WR ... INP  @2  ->        . . . . . . . .   * * * * * * * *
RESET .... INP  @11 ->        * * * * * * * *   . . . . . . . .
WR_CLK ... CKR  @13 ->        * * * * * * * *   . . . . . . . .
REG_CS ... INP  @23 ->        . . . . . . . .   * * * * * * * *
                              C C C C C C C C   D D D D D D D D
                              T T T T T T T T   B B B B B B B B
                              L L L L L L L L   0 1 2 3 4 5 6 7

                              O O O O O O O O
                              U U U U U U U U
                              T T T T T T T T
                              0 1 2 3 4 5 6 7


. = not connected             x = no connection possible
* = signal feeds cell         ? = error, unable to fit

                                                        292074-43
```

Figure 29. iPLS II Report File For The 85C060 Custom Register Design (Continued)

6.0 85C060 PROGRAMMING/ DEVELOPMENT SUPPORT

Design development and device programming support are important issues for PLDs because the silicon is useless without them. Design development tools such as Intel's iPLS II and Data I/O's ABEL are required to convert state machine/boolean equation entries into the required device JEDEC file. The device programmer is then required to program each cell/fuse to configure the device to the user's needs.

Since devices with architectures identical to the 85C060 have been available for a number of years, there is an existing level of support for this PLD already in existence. As mentioned earlier, the architecture, pin-out, and JEDEC map for the 85C060 is identical to the Intel 5C060, Altera EP600, EP610, and EP630, AMD PALCE630, and TI EP610 and EP630. Therefore, performance of existing designs using these devices can be improved by replacing these parts using the existing design/JEDEC files. Any logic compiler (such as ABEL, CUPL, iPLS II) which includes support of these older devices can be used to generate an 85C060 JEDEC file—even if the 85C060 device name is not explicitly supported. Likewise, support is available on most device programming platforms for the 85C060. Table 8 provides a summary of Intel support along with major third-party vendors.

Table 8. 85C060 Development Support Tool Summary

Support Type	Company	Tool Name	85C060 Support	Existing EP6x0, 5C060 Support ?
Programmers	Intel	GUPI (Module Logic IID)	Existing	Yes (same GUPI Module)
	DATA I/O	UNISITE	V3.1	Yes
	DATA I/O	Model 2900	V1.1	Yes (Ver. 1.0)
	DATA I/O	Model 29B	303A-011A 303A-011B	Yes (Ver. 1) Yes (Ver. 1)
	Logical Devices	ALLPRO	V1.49	Yes (Ver 1.46)
Logic Compilers (Design Entry)	Intel	iPLS II	Ver 2.2	Yes (Ver 1.5)
	DATA I/O	ABEL	Ver 3.1	Yes (Ver 3.0)
	Logical Devices	CUPL	Ver 3.3	Yes (Ver 2.15)
Simulation Models	Intel	IPLDView-286	Call for Update	Yes
	Viewlogic	Workview	Call for Update	Yes
	Quadtree/LAI	N/A	Call for Update	Yes

NOTE:
Call the Intel EPLD Hotline for more recent information (1-800-323-EPLD).

The support summary shows an emphasis by Intel to provide timing/simulation models from Viewlogic and Quadtree for customers performing device/board simulations. The models are provided on most popular design platforms. The support summary reflects Intel's close relationship with key support tool vendors (which includes others not shown here) to provide 85C060 customers design support with the format and package with which they are most familiar. These support tool vendors, such as Data I/O and Logical Devices, provide products in addition to the Intel development products. Intel provides complete design development support as detailed in Figure 5.2. Designs can be entered in several ways:

1. Schematic Capture using IPLDview-286—this Viewlogic Workview derivative allows gate/74xx macro level entry and supports design simulation (functional or timing).
2. State Machine entry using iSTATE—this Intel tool translates SMF (State Machine Format) designs into an ADF file.
3. Boolean Equation entry using iPLS II—this logic compiler/minimizer generates device JEDEC/Report files from an ADF source file.

Based on the variety of offerings, users can pick the tool(s) that best fits their development environment. Also, programming support is available with Intel's GUPI adapters, which provide a quick and efficient method of device programming.

7.0 UPGRADING TO THE 85C060

Because of the performance/architecture combination provided by the 85C060 μPLD it may be desirable to upgrade an existing design for many reasons. Upgrading from 5C060 or EP6x0 devices can provide quick performance increases. Upgrading from 22V10 or 20RA10 devices may be done for performance, architecture, heat or power consumption reasons and may include the need for the standby power mode. Conversion of each of the devices mentioned here follows.

UPGRADE FROM 5C060

The Intel 5C060 PLD is pin- and JEDEC-compatible with the 85C060 μPLD. This allows existing 5C060 JEDEC files to be programmed directly into 85C060 μPLDs. Therefore, designs running on 45 ns–55 ns t_{PD} performance levels can instantly be upgraded to 12 ns–25 ns t_{PD} performance levels provided by the 85C060. Intel development tools including iPLS II and GUPI Logic IID programmer adapter support both Intel PLDs. Most non-Intel design development tools and device programmers also support both Intel devices. However, if 85C060 support is not yet available within a specific design tool, the 85C060 design can often be accomplished by referring to the device as a 5C060. In the case of simulation models where specific device timings are used, explicit 85C060 support is required.

UPGRADE FROM ALTERA/TI/AMD EP6x0

As discussed in Section 3, the A.C. and D.C. specifications of the 85C060 meet or exceed those of competitive devices from Altera, TI and AMD. The list of devices which can be upgraded to the Intel 85C060 μPLD includes:

— Altera EP600
— Altera EP610
— Altera EP630
— TI EP610
— TI EP630
— AMD PALCE 630

Just as with the Intel 5C060, an upgrade of one of these devices can be accomplished by using existing JEDEC files from one of these devices to program directly onto an 85C060 μPLD. Also, Data I/O ABEL, Logical Devices CUPL and other design tools support the 85C060, though sometimes under one of the other device names. The only difference of note is the lack of a programmable standby current mode in the AMD device. Again, due to complete architecture compatibility the 85C060 provides a quick and easy performance increase to any design using any of the above devices.

Due to differences in device programming parameters (such as programming voltages and pulse width requirements) all device programmers may not support the 85C060 unless explicitly stated. Also, device simulation models are not interchangeable between devices due to timing differences.

UPGRADE FROM 22V10

Many designers requiring more architectural features than standard PAL devices offer have turned to the 22V10. Compared to standard PALs, the 22V10 offers additional outputs and product terms. However, the 85C060 also has features not found in PAL devices. Since there are many differences between the 85C060 and 22V10, the analysis will revolve first around device pin-out and then, cover performance, architecture, clocking options and D.C. specification issues.

Device Pinout

Table 9 shows the device pinouts of the 85C060 μPLD and the 22V10 and highlights the differences. The 85C060 has dedicated clock inputs at pins 1 and 13 while both of these can be inputs on the 22V10. The 85C060 also has dedicated inputs at pins 14 and 23, while these are I/O pins on the 22V10. However, there

Table 9. 85C060/22V10 Pin-Out Comparison

DIP Pin #	85C060	22V10	Notes
1	CLK1	CLK/INPUT	Clock only on 85C060
2	INPUT	INPUT	
3	I/O	INPUT	Input only on 22V10
4	I/O	INPUT	Input only on 22V10
5	I/O	INPUT	Input only on 22V10
6	I/O	INPUT	Input only on 22V10
7	I/O	INPUT	Input only on 22V10
8	I/O	INPUT	Input only on 22V10
9	I/O	INPUT	Input only on 22V10
10	I/O	INPUT	Input only on 22V10
11	INPUT	INPUT	
12	GND	GND	
13	CLK2	INPUT	Clock only on 85C060
14	INPUT	I/O	Input only on 85C060
15	I/O	I/O	
16	I/O	I/O	
17	I/O	I/O	
18	I/O	I/O	
19	I/O	I/O	
20	I/O	I/O	
21	I/O	I/O	
22	I/O	I/O	
23	INPUT	I/O	Input only on 85C060
24	V_{CC}	V_{CC}	

are 8 input-only pins (3–10) on the 22V10 that are I/O pins on the 85C060. This is the main architectural advantage provided by the 85C060. Table 10 shows a summary of performance (A.C. specifications of the 85C060) compared to the 22V10–15. This includes combinational logic speed (t_{PD}), register performance (t_{CO_1} and t_{SU}), and maximum register speed (F_{MAX}).

ARCHITECTURE

A comparison of architectures shows that the 85C060 has 6 more I/O pins (macrocells) than the 22V10, even though both are 24-pin devices. The main advantages of the 22V10 architecture are higher number of p-terms/macrocell (up to 16 for some macrocells) and capability of up to 22 inputs (compared to 20 for the 85C060).

The number of p-terms (product terms or AND terms) required by any design is often greatly affected by the register type used. The 22V10 offers only D-type flip-flops, while the 85C060 μPLD offers D, T, JK or RS register types. Register type selection is done on a macrocell-by-macrocell basis, so each specific function with the 85C060 can be better optimized. Figure 30 shows an example of how register type selection can affect the number of p-terms required. The example shown is an MSB of a 4-bit counter which requires 5 p-terms using a D-type flip-flop, but only one p-term if a T-type flip-flop is used. By using the register type selection capability of the 85C060 the designer can further minimize logic requirements.

Table 10. 85C060/22V10 Specification Comparison

Area of Comparison	Feature	Intel 85C060-12	Intel 85C060-15	AMD 22V10-15	Lattice GAL 22V10-15
Performance	t_{PD}	12 ns	15 ns	15 ns	15 ns
	F_{CNT}	66 MHz	50 MHz	50 MHz	50 MHz
	F_{MAX}	100 MHz	66 MHz	50 MHz	62.5 MHz
	t_{CO_1}	7 ns	8 ns	10	8 ns
	t_{SU}	8 ns	12 ns	10	12 ns
Architecture	Macrocells	16	16	10	10
	Outputs	16	16	10	10
	Max Inputs	20	20	22	22
	Prog. Security Cell	Yes	Yes	Yes	Yes
	Ave P-Terms/Macrocell	8	8	12	12
	Individual Macrocell OE	Yes	Yes	Yes	Yes
	Reg Clock Options	3	3	1	1
	Indiv. Macrocell Clear	Yes	Yes	No	No
	Invert Control	Yes	Yes	Yes	Yes
	Reg. Power Up State	Low	Low	High	High
	Register Types	D/T/RS/JK	D/T/RS/JK	D	D
D.C. Specifications	I_{CC}(max)	80 mA	80 mA	180 mA	130*
	Power-Down Mode	Yes	Yes	No	No
	Standby Current	<100 μA	<100 μA	N/A	N/A
	I_{OL} (max)	12 mA	12 mA	16 mA	16 mA

*F_{TOGGLE} = 15 MHz

Clocking Options

One other major architectural improvement provided by the 85C060 μPLD is in the area of logic clocking options. Not only does the 85C060 have two dedicated clock inputs (allowing separate synchronous clocking of macrocells), but each macrocell has a p-term dedicated as an asynchronous clock option.

D.C. Specifications

The final area of comparison between these two devices is D.C. specifications. The advantage in maximum supply current (I_{CC}) has benefits in many areas including power supply selection, cooling requirements and system reliability. The 85C060 μPLD also offers a programmable standby current option which allows I_{CC} values to drop to 20 μA (typical) when in the standby mode. This can be useful if working with a "power budget" such as in laptop PC or Microchannel add-in card design. The difference in I_{OL} capabilities is very small (16 mA for the 22V10 versus 12 mA for the 85C060).

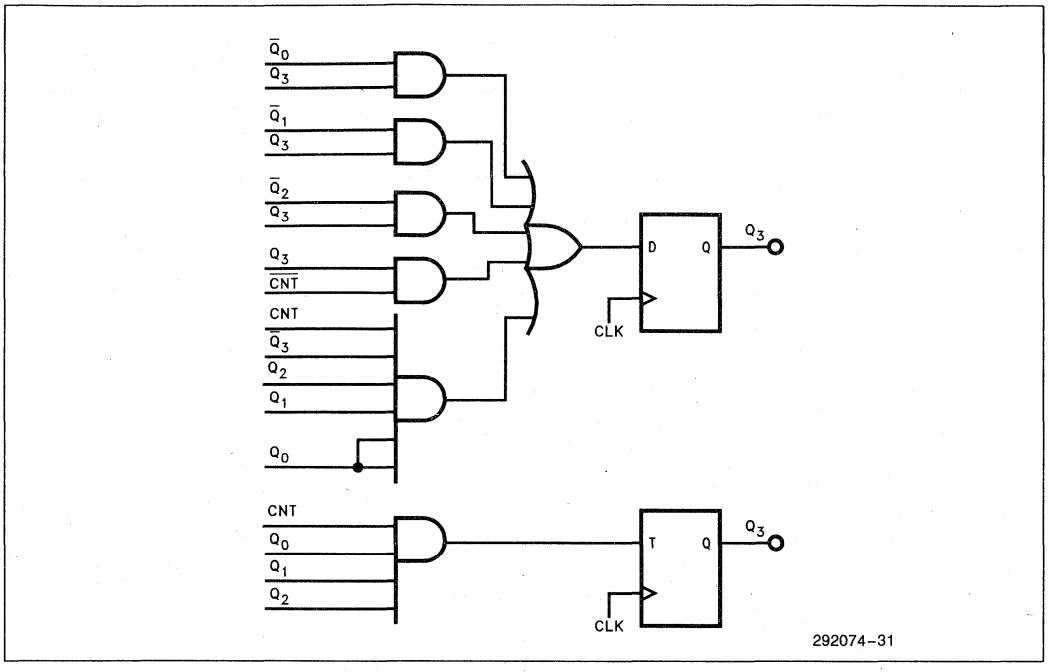


Figure 30. Product Term Requirements for D-Type and T-Type Flip-Flops (MSB of 4-Bit Counter)

22V10 to 85C060 Conversion

NOTE:
There are NO performance related issues for this comparison due to design of 85C060. No issues with registered operation (F_{MAX}, t_{CO}); no issues with combinatorial logic (t_{PD}).

There are several ways to replace a 22V10. A "pin-for-pin" replacement means the 85C060 that is upgrading the 22V10 can be placed into an existing 22V10 socket and provide complete compatibility. A "functional" replacement is one in which the 85C060 can replace the 22V10, but one or more of the pin numbers need to be changed. A functional replacement will only be an issue if the board is already made; if the upgrade is made at the design phase there is very little impact.

1. Determine if a pin for pin replacement can be made:
   a) Are there 8 or less p-terms used for each 22V10 output?
   b) Is pin 1 used as a Clock (rather than an Input)?
   c) Is pin 13 unused (i.e., is this input pin unused on the 22V10)?
   d) Is the global register preset feature unused?
   e) Is the output drive (I_{OL}) requirement for each output 12 mA or less?

NOTE:
Check for need of clock input at Pin 13 of the 85C060.

If YES to all above, then pin-for-pin replacement looks good!

2. Determine if functional replacement can be made:
   a) Is the total number of inputs and outputs 20 or less?
   b) Are there 8 or less p-terms for each output?
      — If NO, can the number be reduced via 85C060 register type selection, inversion control (DeMorgan's logic implementation) and/or use of async clock capabilities?
   c) Is the output drive (I_{OL}) requirement for each output 12 mA or less?
      — If NO, is there a spare 85C060 output which can be utilized to duplicate this output and double the output drive capability?

If YES to questions above, then a functional replacement looks good!

3. In some cases the 5AC312 can also upgrade 22V10 sockets due to it's pin-for-pin compatibility, and superset architecture, although it's performance is not equal to that of the faster 22V10s.

4. In other cases the 22V10 and surrounding PLDs and/or 74xxx logic can be integrated into one of Intel's larger high-performance PLDs (such as the 85C090). This is always an option if there are power, heat, or board space concerns.

> Call The Intel EPLD Hotline for Conversion Assistance
> 1-800-323-EPLD

UPGRADE FROM 20RA10

The 20RA10 presents designers requiring implementation of an asynchronous state machine or other asynchronous registered logic with a viable solution. The Intel 85C060 provides these designers with another alternative. In many cases, a pin-compatible upgrade can be made due to architecture and performance features of the Intel 85C060 μPLD.

Device Pin-Out

Table 11 compares the device pinouts of the 85C060 μPLD and the Lattice 20RA10. Both have up to 20 inputs, both are 24 pin devices (DIP package), and both have dedicated functions implemented at pins 1 and 13. The advantage provided by the 85C060 are the 16 I/Os (macrocells), compared to only 10 for the 20RA10. This allows designers the flexibility to implement larger asynchronous state machines and/or additional synchronous or combinatorial logic.

Spec Comparison

A data sheet comparison of the Intel 85C060 and Lattice 20RA10–15 is summarized in Table 12. The performance of the 85C060 is superior for both combinatorial logic (t_{PD}) and asynchronous register performance (F_{ACNT}, F_{AMAX}, t_{ACO_1}, t_{ASU}) as well as synchronous logic (which is not supported by the 20RA10). There are significant differences in register setup time that could be critical to a designer required to meet specifications of microprocessors or other system peripherals (such as an output valid delay spec).

3

Table 11. 85C060/20RA10 Pin-Out Comparison

DIP Pin #	85C060	22V10	Notes
1	CLK1	PRELOAD	Dedicated on Each
2	INPUT	INPUT	
3	I/O	INPUT	Input only on 20RA10
4	I/O	INPUT	Input only on 20RA10
5	I/O	INPUT	Input only on 20RA10
6	I/O	INPUT	Input only on 20RA10
7	I/O	INPUT	Input only on 20RA10
8	I/O	INPUT	Input only on 20RA10
9	I/O	INPUT	Input only on 20RA10
10	I/O	INPUT	Input only on 20RA10
11	INPUT	INPUT	
12	GND	GND	
13	CLK2	OE	Dedicated on Each
14	INPUT	I/O	Input only on 85C060
15	I/O	I/O	
16	I/O	I/O	
17	I/O	I/O	
18	I/O	I/O	
19	I/O	I/O	
20	I/O	I/O	
21	I/O	I/O	
22	I/O	I/O	
23	INPUT	I/O	Input only on 85C060
24	V_{CC}	V_{CC}	

Table 12. 85C060/20RA10 Specification Comparison Summary

Area of Comparison	Feature	Intel 85C060-12	Lattice 20RA10-15
Performance	t_{PD}	12 ns	15 ns
	F_{ACNT_1}	58.8 MHz	45 MHz
	F_{AMAX}	83.3 MHz	50 MHz
	t_{ACO_1}	14 ns	15 ns
	t_{ASU}	3 ns	7 ns
Architecture	DIP Pin Count	24	24
	Macrocells	16	10
	Outputs (I/Os)	16	10
	Inputs (max)	20	20
	Prog. Security Cell	Yes	No
	Total P-Terms/ Macrocell	10	8
	Indiv. Macrocell OE	Yes	No
	Reg. Preload	No	Yes
	Invert Control	Yes	Yes
	Reg. Types	D/T/JK/RS	D
	Clock Options	Sync/Async	Async Only
	Async Preset	No	Yes
	Async Clear	Yes	Yes
D.C. Specs	I_{CC} (max)	80 mA@66 MHz	100 mA@15 MHz
	Power Down Mode	Yes	No
	Standby I_{CC}	150 μA (max)	N/A
	I_{OL} (max)	12 mA	8 mA

Architecture

Comparing architectures, the 85C060 μPLD offers more flexibility than the 20RA10. This is based on macrocell count, total p-terms/macrocell, register type selections, and clock options. The 85C060 offers 6 more macrocells than the 20RA10. It also has 8 dedicated sum-of-product p-terms in addition to asynchronous register reset (clear) and async clock/OE p-terms. The 20RA10 has a total of eight p-terms for each macrocell, only four of which are dedicated to performing standard sum-of-products logic. Register type selections offered by the 85C060 include D, T, RS and JK, while the 20RA10 implements only a D-type register. As mentioned in the section on upgrading from the 22V10, register type selection can have a significant impact on the number of p-terms required to implement a logic function and may allow a design to "fit" with one type, but not with another. Also, of note is the fact that the 85C060 can implement synchronous (via 2 clock inputs) or asynchronous (via individual macrocell p-terms) registered logic. The designer using the 20RA10 can only implement asynchronous registered logic (using clock p-term) only. As mentioned earlier the synchronous register specifications of the 85C060 (t_{SU}, t_H and t_{CO_1}) are slightly skewed from the asynchronous register specifications (t_{ASU}, t_{AH} and t_{ACO_1}). A designer using the 85C060 can select clocking options on a macrocell-by-macrocell basis to best meet the needs of the system.

D.C. Specifications

Looking at D.C. specifications of these devices the advantages of the 85C060 are clear. The 20RA10 consumes up to 100 mA at 15 MHz operation, while the Intel 85C060 μPLD consumes only 80 mA at 66 MHz operation. A typical I_{CC} value for the 85C060 at 15 MHz would be less than 70 mA. In addition the 85C060 offers a programmable power-down mode, which can put this device into a power saving mode useful in many applications. The I_{OL} of the 85C060 is higher than the 8 mA maximum for the 20RA10.

8.0 SUMMARY

The 85C060 represents the highest performer in one of the industry-standard PLD architectures. The advantages gained by using the 85C060 extend past the obvi-

ous performance advantages of a 12 ns propagation delay and 66 MHz counter capability. The 85C060 μPLD provides the following capabilities:

- 12 mA I_{OL}
- Programmable Standby Mode
- 80 mA (max) I_{CC}
- 16 Macrocell Architecture
- 100% Testability due to EPROM Technology
- 100 MHz Register Operation

In addition, this device is offered with several speed and package options. The device is available in 12 ns, 15 ns and 25 ns t_{PD} speeds. The packages for the 85C060 μPLD are Ceramic DIP, Plastic DIP and PLCC.

Already highlighted were the architectural advantages of this device over standard programmable logic devices. These advantages include device level capabilities (16 macrocells, up to 20 inputs, dual clocking) and macrocell level capabilities (total of 10 p-terms per macrocell, register type selection, feedback control, async clocking). These capabilities of the 85C060 compared to competitive devices showed clearly this device outperforms others at a data sheet level (A.C. and D.C. specifications). But, the 85C060 was further highlighted by the characterization data presented, which demonstrates performance outside data sheet considerations. The combination of data sheet and device characterization information provides designers an added level of confidence in actual device behavior, and knowledge of actual in-system performance.

The 85C060 device is well supported by PLD design and development tools. There are many existing tools that support this device as a 5C060, EP600, etc., but, specific 85C060 tools are being completed.

One other advantage of the 85C060 μPLD is the ease of upgrade it provides to systems using the 5C060, EP610/630, 22V10 or 20RA10. These devices can be quickly upgraded to the 85C060 μPLD to gain performance, architectural, and power dissipation advantages.

The 85C060 μPLD takes this standard architecture to a higher level. Its high performance in all areas open new applications to Intel μPLDs. Designers can now have both integration and performance!

intel®

APPLICATION NOTE

AP-317

June 1988

Implementing a PS/2 POS Using the 5AC312 EPLD

PEDRO VARGAS
PROGRAMMABLE LOGIC APPLICATIONS

Order Number: 292047-001

IMPLEMENTING A PS/2 POS USING THE 5AC312 EPLD

CONTENTS

*IBM, Personal System/2 and Micro Channel are trademarks of International Business Machines Corporation.

INTRODUCTION

The introduction of the IBM* PS/2 (Personal System/2*) models and the innovative Micro Channel* has provided numerous opportunities to develop creative interface solutions. Although the interface requirements are new, the designer is faced with making a familiar choice: Use discrete chips (SSI/MSI), incorporate a PLD, or go for the custom IC solution.

In the past, using TTL on the PC/XT/AT bus was often a good choice, but the reduced size of the PS/2 adapters ("plug in boards") increases the cost of board space dramatically. The custom chip solution is probably the best for companies that have a well-defined product, large volumes, and can afford the cost of the chip development. The third choice, using a PLD, is one that has not been popular in PC bus interfacing due to the limited function and performance of most PLDs.

The Intel 5AC312 is a third-generation EPLD that gives designers the resources needed to interface to buses like the Micro Channel. In addition, it provides two benefits not completely provided by either of the other two choices; high integration, and re-programmability. The rest of this application note contains a detailed presentation of a basic POS (Programmable Option Select) implementation for the PS/2 Micro Channel that is done with the 5AC312 EPLD.

PS/2 MICRO CHANNEL

One of the best features in the PS/2 models is the capability to do system and adapter configuration with software instead of hardware. This feature, called POS (Programmable Option Select), eliminates the need for switches on the motherboard and adapters by replacing them with programmable registers. The idea is, rather than removing boards and manually setting switches, all configuration information is located in files and can be read or written to the motherboard or to the adapters through the Micro Channel. The motherboard and each connector on the Micro Channel has a unique signal called -CD SETUP that initiates a setup mode when it is active. Only one connector at a time can be in the setup mode, which provides an organized way to perform initialization.

POS REQUIREMENTS

Each adapter must implement POS with eight registers. Depending on the adapter function, not all of them need to be used. The first three (POS registers 0,1,2) are required because they provide the adapter ID and the adapter enable/disable function necessary during setup and error checking. In brief, the way that the system uses POS is as follows:

1. The system selects the adapter to be placed in setup mode by driving its -CD SETUP signal active.
2. The adapter is identified by reading two ID bytes from POS 0 and POS 1 (HEX 100 and 101).
3. The adapter is disabled by writing "0" to POS 2 (HEX 102).
4. If implemented, Option Select Data is written to POS 3, 4, 5.
5. The adapter is enabled by writing "1" to POS 2.
6. The adapter is out of setup mode when the system drives the -CD SETUP signal inactive.

The actual hardware implementation of POS is summarized in IBM technical documents, but the details are left up to each designer.

ADAPTER REQUIREMENTS

The adapter used for this design is an Intel single-function card that incorporates two modems controlled by a 80C186. Since it performs only one function, there was no need to implement the POS Option Select bytes. (These POS bytes are used with multi- function adapters that do more than one task and reside in the system with similar adapters.) In this case, the only requirements were to provide the ID bytes and the enable/disable features, which are done with POS registers 0,1, and 2. Figure 1 shows the POS register layout and the typical POS hardware implementation as suggested by IBM. Table 1 defines the POS registers.

*IBM, Personal System/2 and Micro Channel are trademarks of International Business Machines Corporation.

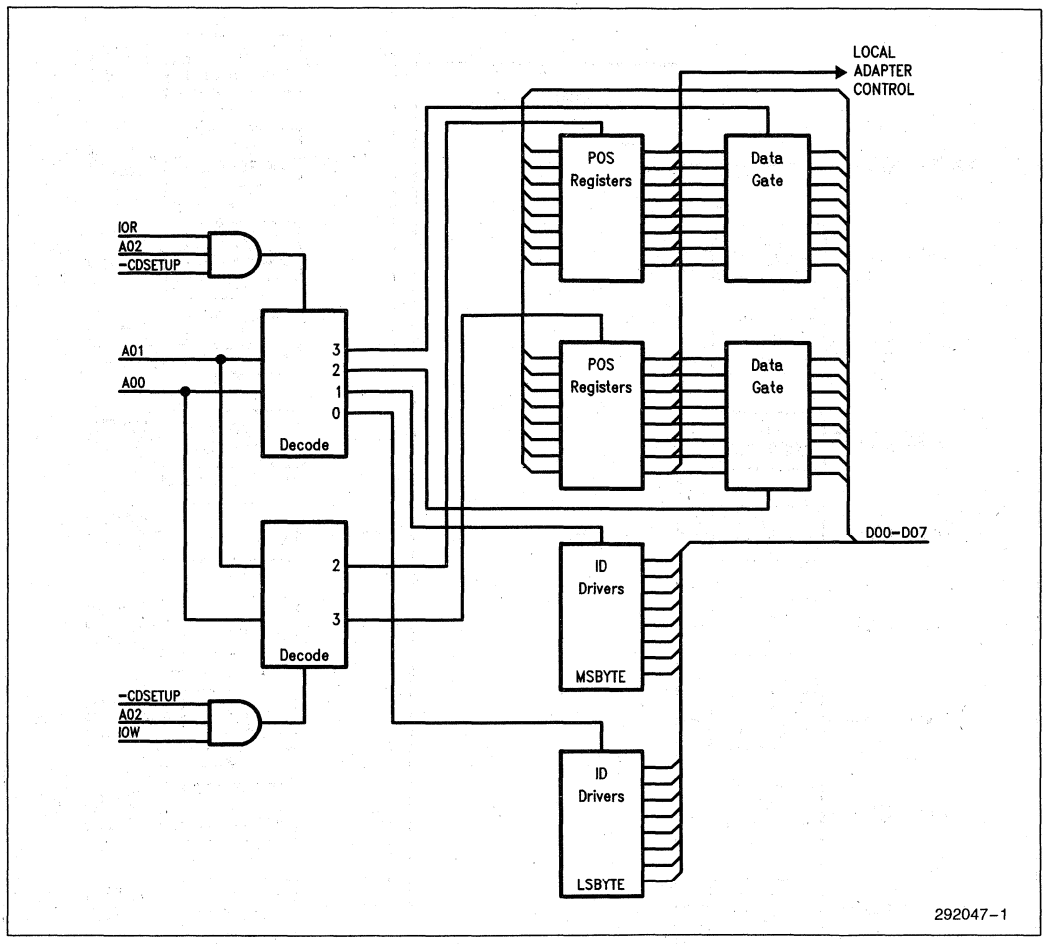


Figure 1. Typical Adaptor Implementation of POS

Table 1. POS I/O Address Decode

Address (hex)	Register	-CD SETUP	Address Bit			Function
			A2	A1	A0	
0100	POS Register 0	0	0	0	0	Adapter Identification Byte (Least Significant Byte)
0101	POS Register 1	0	0	0	1	Adapter Identification Byte (Most Significant Byte)
0102	POS Register 2	0	0	1	0	Option Select Data (Byte 1)*
0103	POS Register 3	0	0	1	1	Option Select Data (Byte 2)
0104	POS Register 4	0	1	0	0	Option Select Data (Byte 3)
0105	POS Register 5	0	1	0	1	Option Select Data (Byte 4)*
0106	POS Register 6	0	1	1	0	Subaddress Extension (Least Significant Byte)
0107	POS Register 7	0	1	1	1	Subaddress Extension (Most Significant Byte)

*These bytes contain one or more bits with specific value assignments

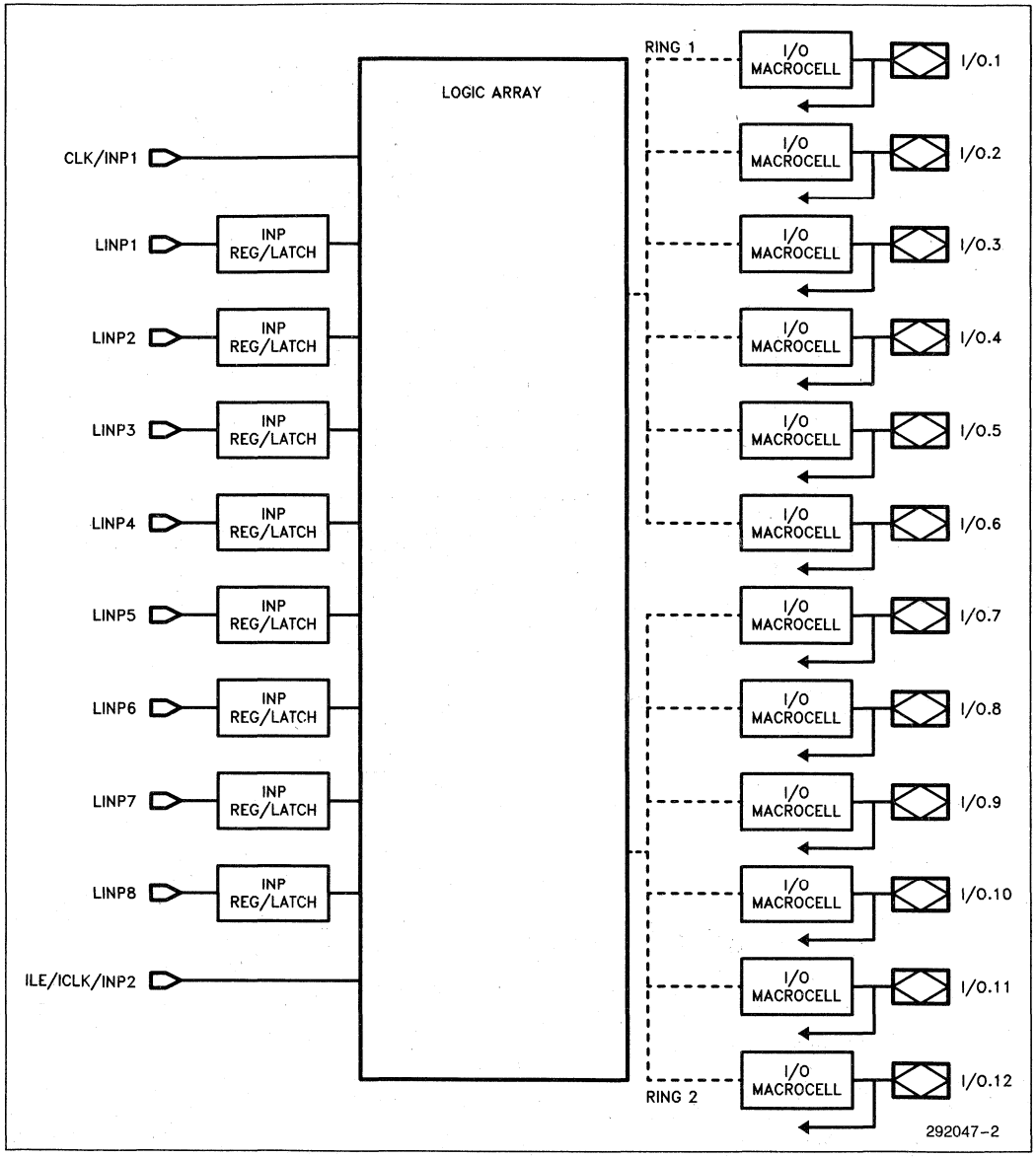


Figure 2. 5AC312 Architecture

5AC312 EPLD

With 12 macrocells and a host of other features, the 5AC312 is Intel's newest EPLD. The device is based on the same CHMOS process used in other Intel devices. This EPLD provides an abundant feature set, but its strength lies in being able to efficiently implement one very important function missing from most PLDs: register-logic-register functions.

DEVICE DESCRIPTION

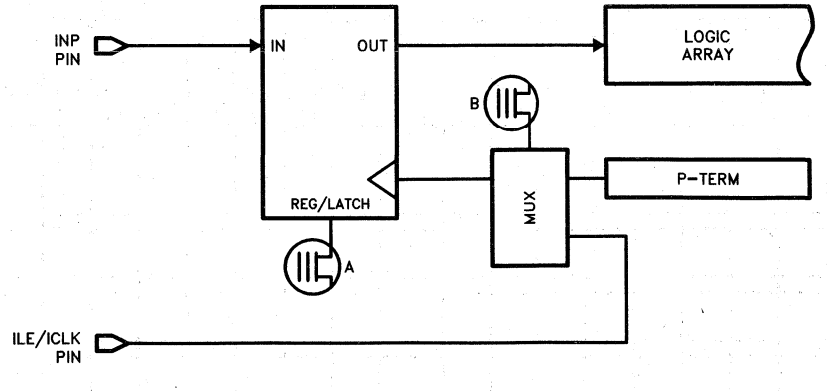
The 5AC312 (Figure 2) contains 12 macrocells with programmable outputs and inputs. A macrocell is the basic block associated with each output register within the EPLD. The 5AC312 has the following features:

- 12 I/O macrocells with dual feedback for implementing buried registers.

- 8 programmable inputs that can be configured as latches, registers, or flow through inputs. These can be clocked synchronously or asynchronously.
- Product term allocation on each macrocell.
- 2 product terms on all macrocell control signals.
- 2 multi-function pins; a CLK/INPUT and a ILE/ICLK/INPUT.
- 40 MHz operation.

The 5AC312 provides three major benefits that are especially important to designers working on bus interfaces:

1. The availability of input latches (Figure 3) makes it easy to synchronize bus control signals synchronously or asynchronously. The latches can be clocked as a group of 8 or individually, as is quite common on most buses. Input latches also make state machine designs more reliable. Since buses are prone to glitches and other transients, the ability to hold the inputs stable while transitioning through states makes the difference between a clean and a jittery state machine.
2. Product Term Allocation (Patent Pending) brings a new concept to the Intel EPLD family and makes the 5AC312 unique among PLDs. This feature means that the designer can implement large designs that contain as few as zero or as many as 16 product terms per macrocell (Figure 4). Product Term Allocation takes place in two rings of six macrocells. Within each ring (Figure 2), individual macrocells can allocate p-terms to/from adjacent macrocells. This is a real benefit in bus decoding where intermediate signals can have few or many p-terms all within the same logic function. Most designers that use PLDs have at least one horror story of a design that required 10 or more p-terms and a device that could only provide 8.
3. A flexible output structure is a must for efficient bus interfacing, which quite commonly requires lots of I/O and complex control signals. The 5AC312 meets these demands head-on with dual-feedback paths and two p-terms per control signal on all I/O macrocells. This means that certain functions, like state machines, can be buried and a pin won't be wasted because it can be used as an additional input. Also, output enables and register operations are frequently generated by a combination of memory, I/O, read, and write strobes. Many times these control signals require two p-terms or the equivalent of an external read/write multiplexer. Prior to the 5AC312, the only way to implement this in PLDs was to waste a macrocell to inefficiently provide this function. Figure 5 shows the macrocell structure and details this third benefit.



NOTE:
Flow-through input selected by connecting ILE P-Term to V_{CC}.

Figure 3. 5AC312 Input Structure

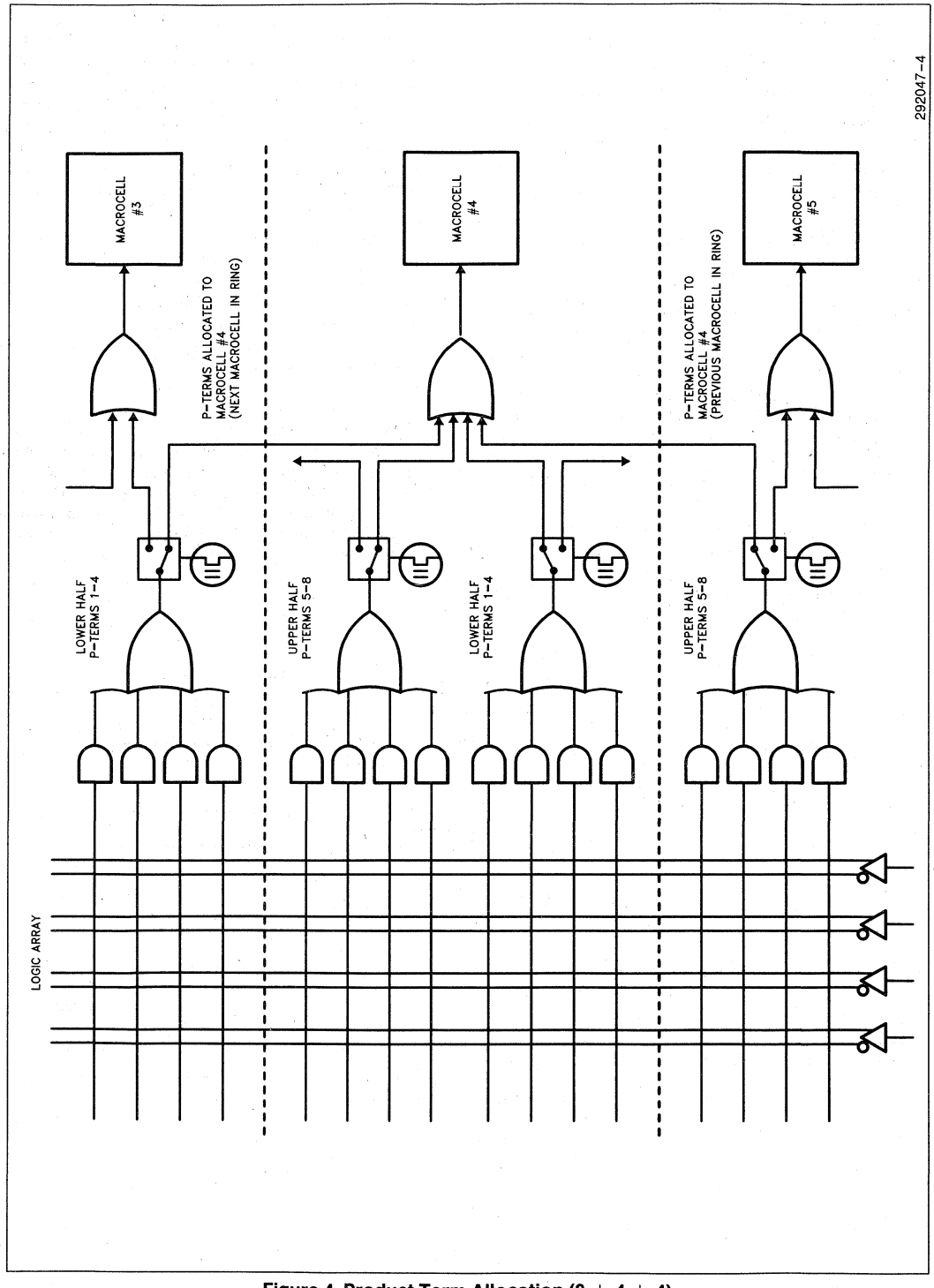


Figure 4. Product Term Allocation (8 + 4 + 4)

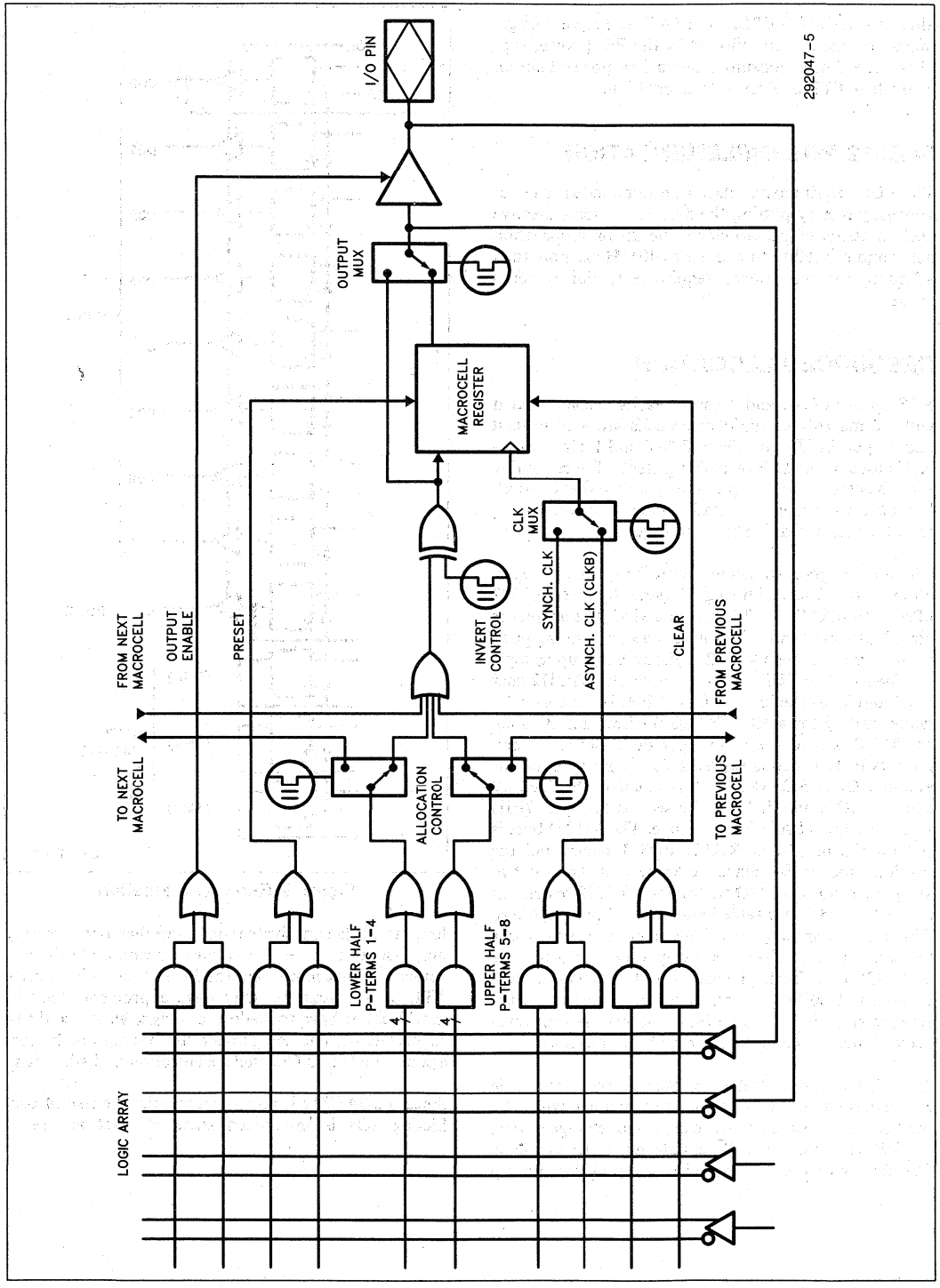


Figure 5. 5AC312 Basic Macrocell Structure

Since it is a CMOS EPLD, the 5AC312 has very frugal impact on the current allotted by the PS/2 power supplies. The device consumes much less power than an equivalent TTL or PAL implementation.

5AC312 POS IMPLEMENTATION

The POS requirements placed on our modem adapter are easily met by putting the 5AC312 to some creative use. In terms of performance, the 25 ns propagation delay and capability to operate to 40MHz is more than adequate for the system requirements during setup mode.

RESOURCE ALLOCATION

POS registers 0, 1, and 2 can be easily accommodated with 12 macrocells. Eight macrocells are used to load and output the ID bytes from POS 0 and 1. One macrocell is used as the LSB of POS register 2. The remaining three macrocells make up a state machine that internally sequences through the setup mode. The partitioning used for this design is shown in Figure 6.

The state diagram shown in Figure 7 explains the operation of the design. During S0 (state 0), the 5AC312 idles until -CD SETUP is driven active. The adapter is already disabled and POS 2 is zero because during power-up and reset the 5AC312 registers come up as logic 0. When -CD SETUP is driven active, the 5AC312 goes to S1 and loads the first ID byte, FFH. It remains in S1 while waiting for a READ POS 0 command. As soon as POS 0 is read the state machine cycles to S2 and outputs FFH which is the least significant byte for our adapter. Once READ POS 0 is inactive, the 5AC312 cycles to S3 where it loads the second ID byte (7FH) and waits for READ POS 1 active. The last ID byte is put on the bus when READ POS 1 comes and the machine goes to S4. Since we know that the next two setup operations are I/O writes, the 5AC312 remains in S5 while POS 2 is disabled and enabled per the Micro Channel bus specification. This operation, which is a bit write of a register, is easily done by using the 5AC312's dual feedback capability. While bit 0 of POS 2 uses the D00 line, internally it gets routed to a separate register. Without dual feedback this internal transceiver function would be impossible to implement.

The IBM Technical Reference Manual provides a table for suggested ID bytes arranged by adapter type. The modem card falls under the category of storage device, so 7FFFH was chosen. While IBM has assigned unique IDs for its own cards the third party choices are up

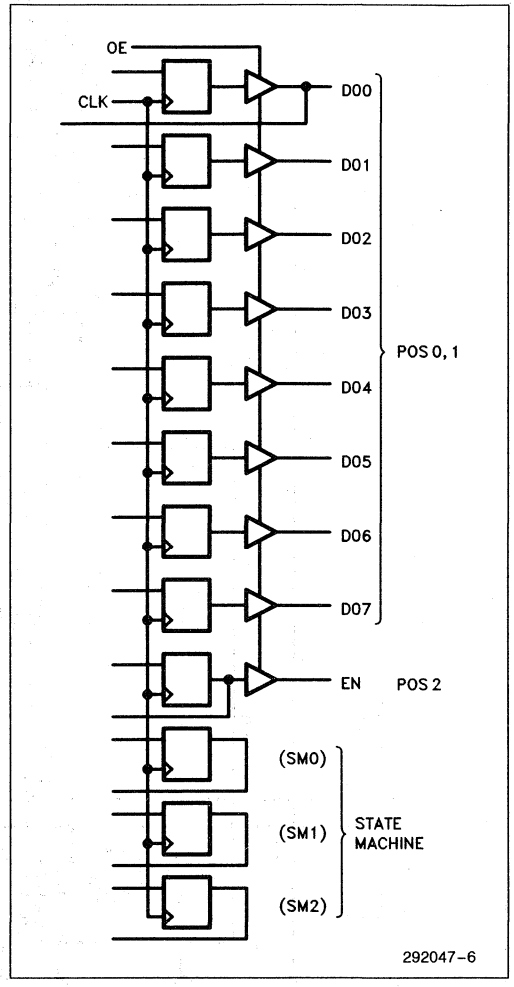


Figure 6. Register Allocation

for grabs. It is conceivable that more than one company may assign the same ID to their own cards. In this case, companies that implement the POS function in discrete TTL or a custom IC may have a problem. That is, they'll either have to re-do the design, which could be expensive, or, cut and jumper the board, which, goes against the Micro Channel specification and still costs.

Since the 5AC312 is re-programmable, the risk of conflicting IDs is minimized since all that is needed

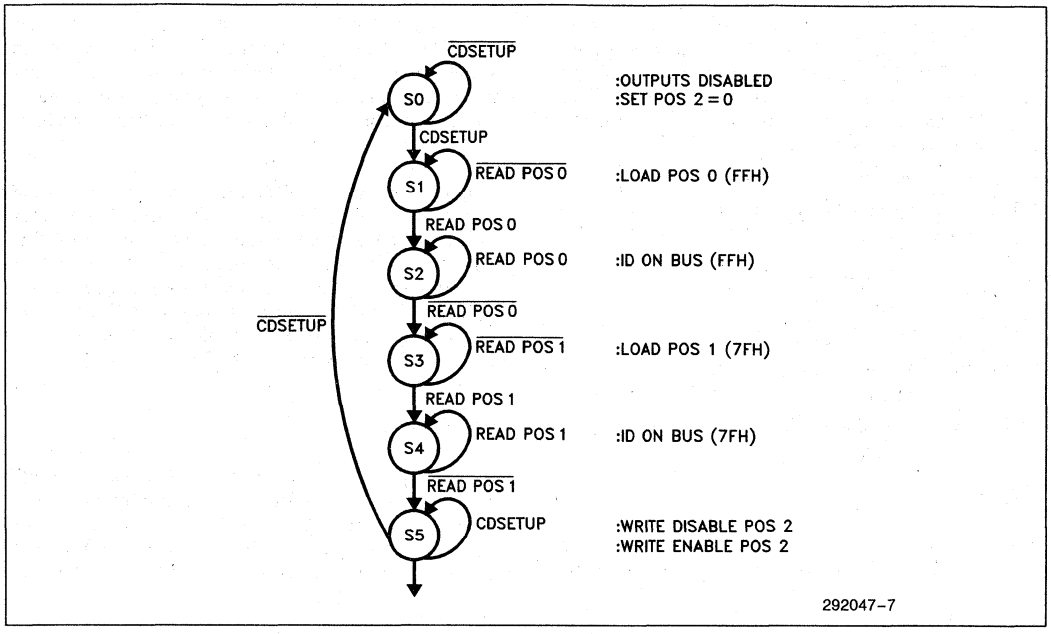


Figure 7. State Diagram

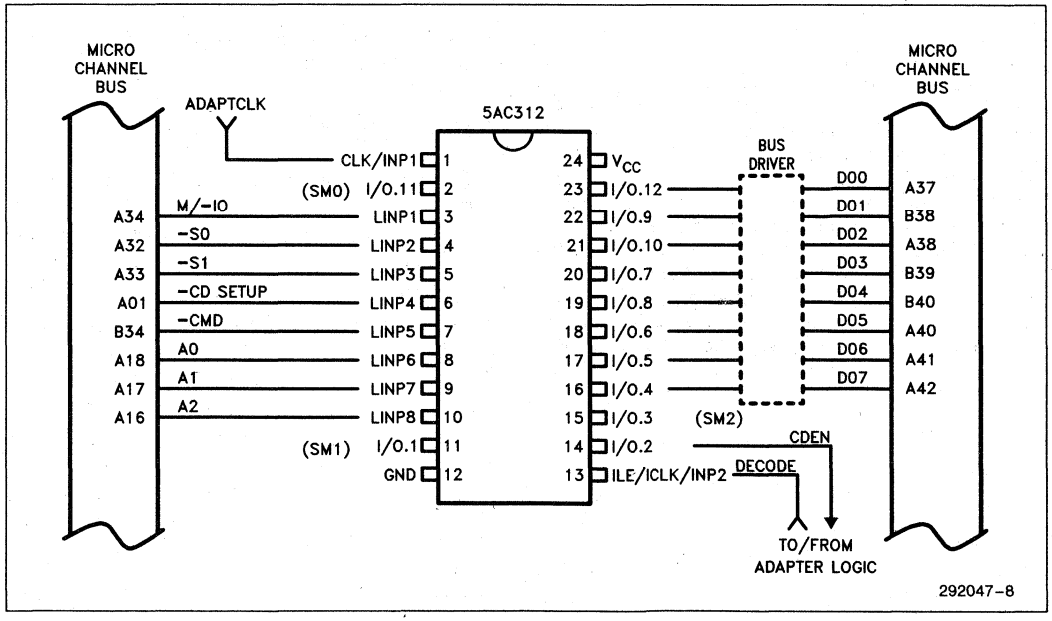


Figure 8. POS Pinout

is to burn another EPLD with the appropriate bytes. The pinout for the 5AC312 POS implementation is shown in Figure 8. Note that a bus driver is required to support the current levels on the Micro Channel. The definition for the signals can be found in the IBM literature, but briefly is as follows:

M/-I/O	Memory or Input/Output.
-S0, -S1	Status bits 0 and 1.
-CD SETUP	Card Setup.
-CMD	Command.
A0–A3	Address bits 0–3.
D00–D07	Eight bit data bus.
DECODE	Adapter decode for the higher order 16 bit address bus.
EN/-DIS	Enable/Disable adapter.

DESIGN FILES

The 5AC312 was developed and programmed using the Intel IPLSII 1.5 EPLD software. This software provides a variety of entry methods like Boolean equation, state machine, and schematic capture. For this particular design, equation entry was the most convenient since the implementation was done in one IC.

The minimized and ordered .LEF (Logic Equation File) for the 5AC312 is shown in Figure 9. A quick glance at a few items really points out the power of the 5AC312. For example, the equation for variable ENd has 11 p-terms. Without p-term allocation this would not have fit unless it was done with two macrocells, which is wasteful. Also, the output enable control signal OE contains two p-terms for each macrocell. Again, without the 5AC312 some work-around may have been possible. But its unlikely that the design could have fit in one device, resulting in a functional but not too optimal solution. Finally, in keeping with good state machine design practices, all of the critical bus signals were received by registers with the EPLD primitive RINP (Registered Input).

A description of the Micro Channel Adapter Description File and the Configuration Utilities is beyond the scope of this article. The IBM literature is very descriptive in how adapters are setup and configured, and the reference section contains the names of the pertinent documents.

This modem adapter POS implementation was easily done in one 5AC312 device. Adding other Micro Channel interface features like arbitration or interrupt sharing would overburden it. Since this adapter did not have those requirements it was not a problem. However, a complete POS implementation with Option Select Bytes and other features could be done with the 5AC312's big brother, the 5AC324.

SUMMARY

Interfacing to the PS/2 Micro Channel can be a difficult chore when using PLDs or other solutions that are not flexible or powerful enough. However, the 5AC312 with its powerful features like product term allocation is a giant step in the right direction to making the job easier.

ACKNOWLEDGEMENTS

Many thanks to Thom Bowns and Dan Smith for their help with this article.

REFERENCES

1. IBM Personal System/2, Model 80, Technical Reference.
2. IBM Personal System/2, Hardware Maintenance Reference.
3. Intel 5AC312 EPLD Data Sheet.
4. *Electronics* "Inside Technology", September 17 1987.

```
THOM BOWNS
INTEL
DECEMBER 4, 1987
1
001
5AC312
Implements POS for the PS/2 using a 5AC312.
LEF Version 1.5 Baseline 4.1i 21 Nov 1987
OPTIONS: TURBO=ON
PART:
      5AC312
INPUTS:
      MIO@3, nS0@4, nS1@5, nCDSETUP@6, nCMD@7, A0@8, A1@9, A2@10, DECODE@13,
      CLK@1, D0@23
OUTPUTS:
      D00@23, D01@22, D02@21, D03@20, D04@19, D05@18, D06@17, D07@16, EN@14,
      SM0@2, SM1@11, SM2@15
NETWORK:
      IRE = CLKB(CLK)
      CLK = INP(CLK)
      MIO = RINP(MIO, IRE, GND, GND)
      nS0 = RINP(nS0, IRE, GND, GND)
      nS1 = RINP(nS1, IRE, GND, GND)
      nCDSETUP = RINP(nCDSETUP, IRE, GND, GND)
      nCMD = RINP(nCMD, IRE, GND, GND)
      A0 = RINP(A0, IRE, GND, GND)
      A1 = RINP(A1, IRE, GND, GND)
      A2 = RINP(A2, IRE, GND, GND)
      DECODE = INP(DECODE)
      D0 = INP(D0)
      D00 = RONF(D00d, CLK, GND, GND, OE)
      D01 = RONF(D01d, CLK, GND, GND, OE)
      D02 = RONF(D02d, CLK, GND, GND, OE)
      D03 = RONF(D03d, CLK, GND, GND, OE)
      D04 = RONF(D04d, CLK, GND, GND, OE)
      D05 = RONF(D05d, CLK, GND, GND, OE)
      D06 = RONF(D06d, CLK, GND, GND, OE)
      D07 = RONF(D07d, CLK, GND, GND, OE)
      EN, EN = RORF(ENd, CLK, GND, GND, VCC)
      SM0 = NORF(SM0d, CLK, GND, GND)
      SM1 = NORF(SM1d, CLK, GND, GND)
      SM2 = NORF(SM2d, CLK, GND, GND)
```

292047-9

Figure 9. POS Design File

```
EQUATIONS:
      SM2d = SM2' * SM1 * SM0 * nCDSETUP' * MIO' * nS0' * A2' * A1' * DECODE *
             nS1
           + SM2 * SM1' * nCDSETUP';

      SM1d = SM2' * SM1' * SM0 * nCDSETUP' * MIO' * nS0' * A2' * A1' * DECODE *
             nS1
           + SM2' * SM1 * SM0' * nCDSETUP'
           + SM2' * SM1 * nCDSETUP' * DECODE'
           + SM2' * SM1 * nCDSETUP' * A1
           + SM2' * SM1 * nCDSETUP' * A2
           + SM2' * SM1 * nCDSETUP' * nS1'
           + SM2' * SM1 * nCDSETUP' * nS0
           + SM2' * SM1 * nCDSETUP' * MIO;

      SM0d = (SM2' * SM0 * MIO' * nS0' * nS1 * A2' * A1' * DECODE
           +  SM2 * SM0' * MIO' * nS0' * nS1 * A2' * A1' * DECODE
           +  SM1 * MIO' * nS0' * nS1 * A2' * A1' * DECODE
           +  SM2 * SM1
           +  nCDSETUP)';

      ENd = D0 * MIO' * A2' * A1 * A0' * DECODE * SM2 * SM1' * SM0 * nS1' * nS0
          + nS0' * EN
          + nS1 * EN
          + SM0' * EN
          + SM1 * EN
          + SM2' * EN
          + DECODE' * EN
          + A0 * EN
          + A1' * EN
          + A2 * EN
          + MIO * EN;

      D07d = SM2' * SM0'
           + SM2' * SM1';

      D06d = (SM2 * SM1)';

      D05d = (SM2 * SM1)';

      D04d = (SM2 * SM1)';

      D03d = (SM2 * SM1)';

      D02d = (SM2 * SM1)';

      D01d = (SM2 * SM1)';

      OE = SM2 * SM1' * SM0' * MIO' * nS0' * A2' * A1' * DECODE * nS1
         + SM2' * SM1 * SM0' * MIO' * nS0' * A2' * A1' * DECODE * nS1;

      D00d = (SM2 * SM1)';

END$
```

292047-10

Figure 9. POS Design File (Continued)

December 1988

Designing with the 5AC312/5AC324 EPLDs

DAVID BICKEL
PROGRAMMABLE LOGIC APPLICATIONS
INTEL CORPORATION

Order Number: 292049-001

DESIGNING WITH THE 5AC312/5AC324 EPLDS

CONTENTS

INTRODUCTION

The Intel 5AC312 EPLD (Erasable Programmable Logic Device) was developed to break down certain existing PLD architectural barriers and meet increased performance needs. The Intel 5AC312 EPLD was designed by EPLD users with direct input from system designers. In the design process, emphasis was placed first on gate utilization, and then on density.

This application note highlights the advanced architecture and features of the 5AC312 EPLD and shows the benefits of designing with this new device over more traditional PLD architectures. These features include enhanced input structure with register/latch option on all input pins (synchronous or asynchronous operation); user-controllable, software-supported p-term allocation scheme in all macrocells; and multiple p-terms on control functions (asynchronous CLK, SET, RESET, OE).

It should also be noted that the features and information described here also apply to the new 5AC324 EPLD. The 5AC324 is basically a 24 macrocell version of the 5AC312.

PROGRAMMABLE INPUTS

The 5AC312 was designed with a highly flexible macrocell and I/O structure allowing the device to implement both combinational and sequential logic functions. The enhanced input structure not only allows the device to latch and hold incoming data, but also to implement register-combinational-register logic to easily accommodate state machine designs. Figure 1 shows a global view of the 5AC312 architecture.

The 5AC312 is equipped with 8 user-programmable input structures that can each be configured to work in one of five modes: 1) synchronous D-type register, 2) asynchronous D-type register, 3) synchronous D-type latch, 4) asynchronous D-type latch, and 5) flow-through input. Each input can be configured independently of the others. The desired configuration is implemented through the programming of EPROM architecture control bits by the logic compiler under user-control.

MACROCELL STRUCTURE

The 5AC312 also has a unique macrocell array structure that allows for user-controllable, software-supported product term allocation in each of its 12 macrocells. Each of the 12 macrocells also has a dual feedback option with independent feedback and I/O paths. Each macrocell has 16 product terms, 8 of which control the OE, PRESET, ASYNCHRONOUS CLOCK, and CLEAR signals (2 p-terms per signal). The other 8 feed the data input to the macrocell and are split into two groups of four (upper half and lower half). See Figure 2. Each group of four can be allocated to an adjacent macrocell if needed.

As shown in Figure 1, the 12 macrocells of the 5AC312 are further divided into two "rings" with 6 macrocells per ring. Allocation of p-terms to adjacent macrocells can occur with a given ring. See Figure 3 for p-term allocation scheme.

Each macrocell register in the 5AC312 is also equipped with an asynchronous PRESET signal. The PRESET function can be constructed in more traditional architectural devices such as the 5C060 and 5C090 using combinational logic and feedback, however two macrocells are consumed in the process. To illustrate this difference, compare Figure 2 to the implementation shown in Figure 4. The PRESET function would require additional macrocells in traditional architectures if it were expanded beyond a single p-term.

MULTIPLE P-TERMS

Multiple p-terms on the control functions (asynchronous CLOCK, PRESET, RESET, and OE) increases the efficiency of the device. Multiplexed I/O is accomplished by controlling the output buffer associated with each macrocell using the 2 p-terms set aside for implementing an OE function. Multiple p-terms create a means to avoid using macrocells for control logic. For example, it would take two macrocells in the 5C060 and 5C090 EPLD to drive the OE line by a 2 p-term signal. To illustrate, compare Figure 2, the 5AC312 macrocell structure, to Figure 5, a diagram of how a two p-term OE signal can be implemented in a 5C060 or 5C090 EPLD.

P-TERM ALLOCATION

P-term allocation allows for more efficient use of p-terms and thus increased device utilization by raising the number of p-terms per macrocell to 16. P-term allocation, where p-terms are dedicated to certain macrocells, should not be confused with p-term sharing, where several macrocells can actually use the same p-terms. The p-term allocation scheme in all macrocells is user-controllable and software supported, and provides the ability to satisfy designs with large p-term requirements. P-term allocation is ideal for p-term intensive applications such as complex counters or comparators.

P-term allocation in the 5AC312 is used when a design requires one of the 12 macrocells to employ more than 8 p-terms. P-term allocation is simply the transfer

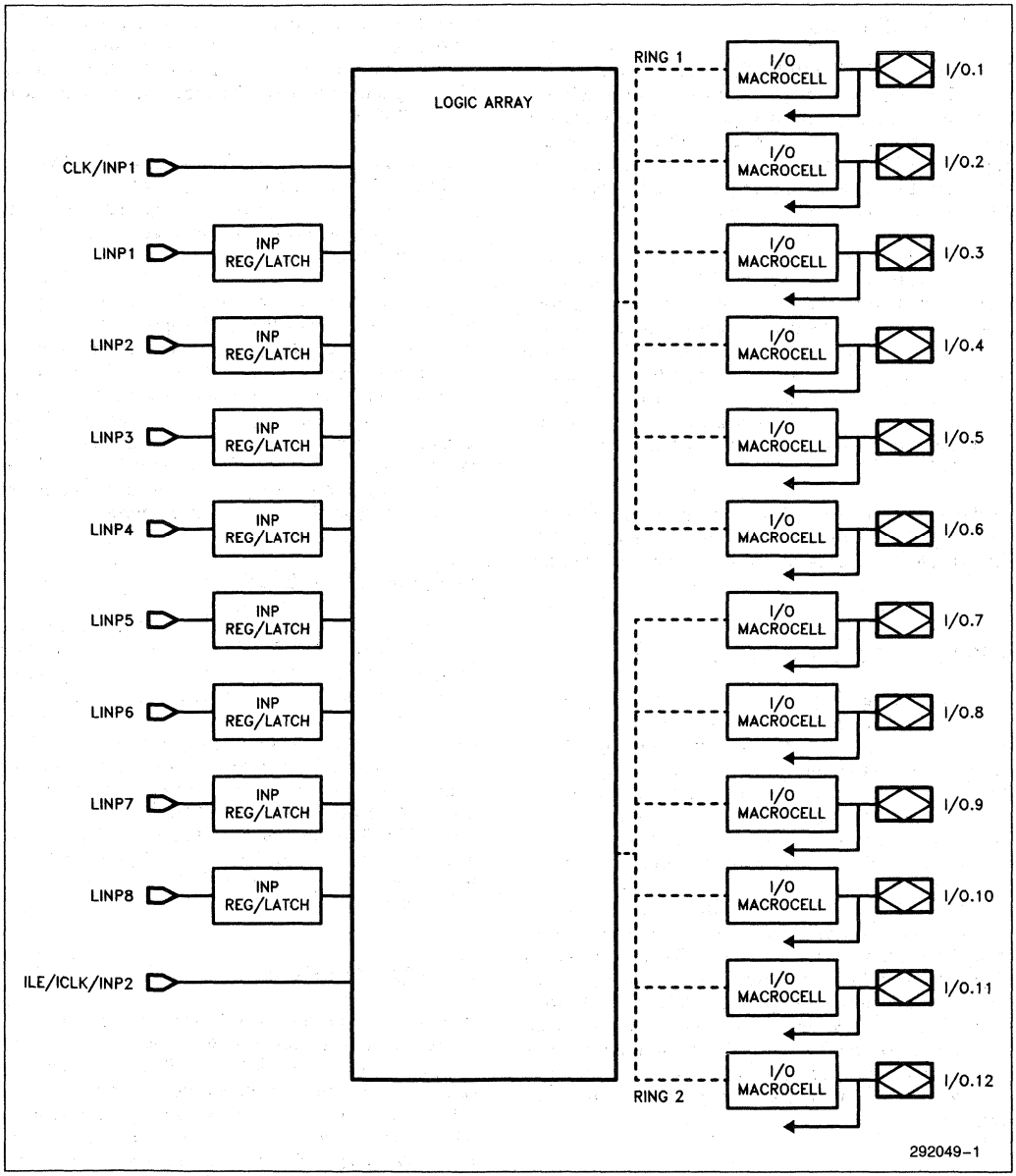


Figure 1. 5AC312 Architecture

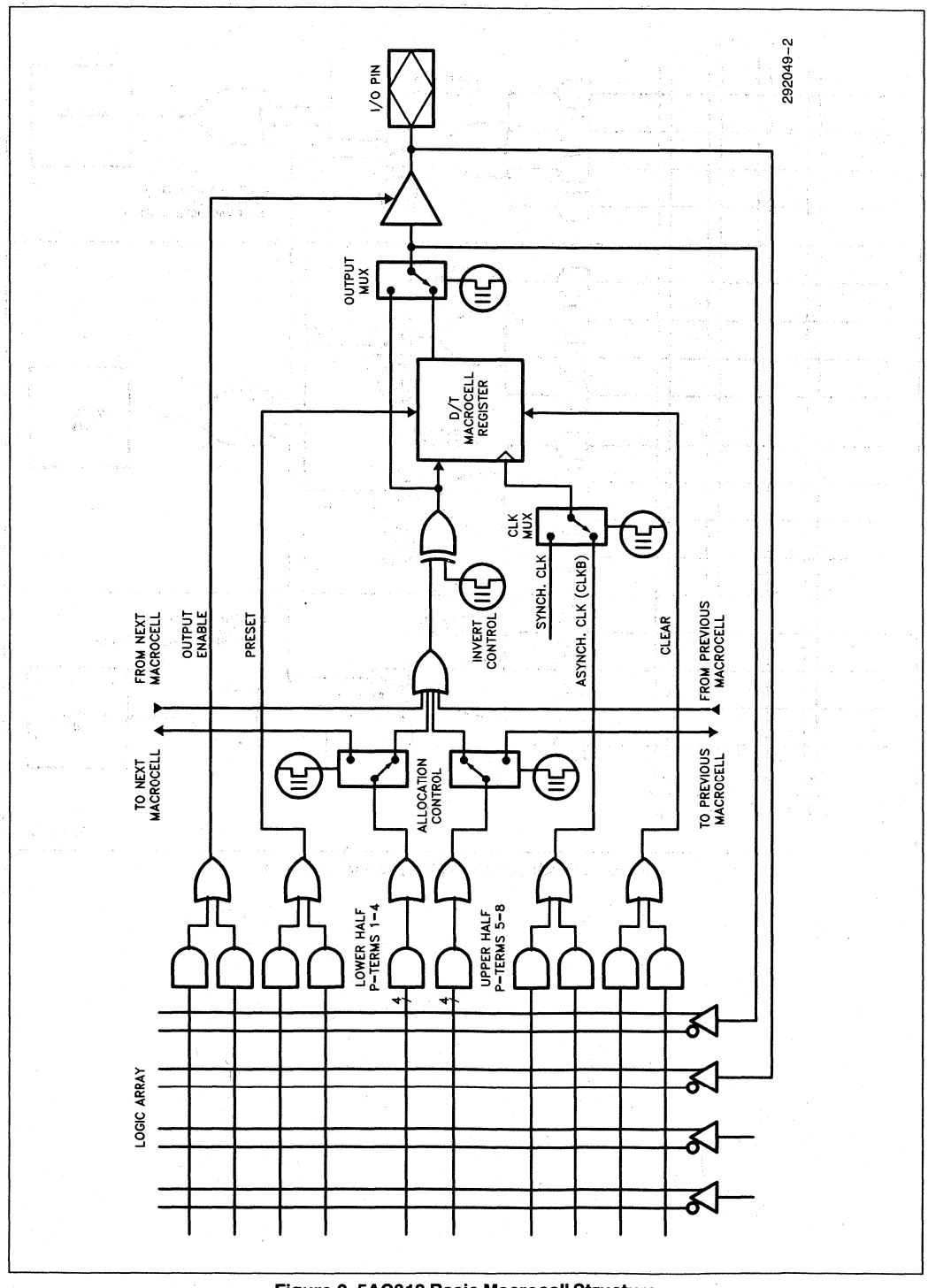


Figure 2. 5AC312 Basic Macrocell Structure

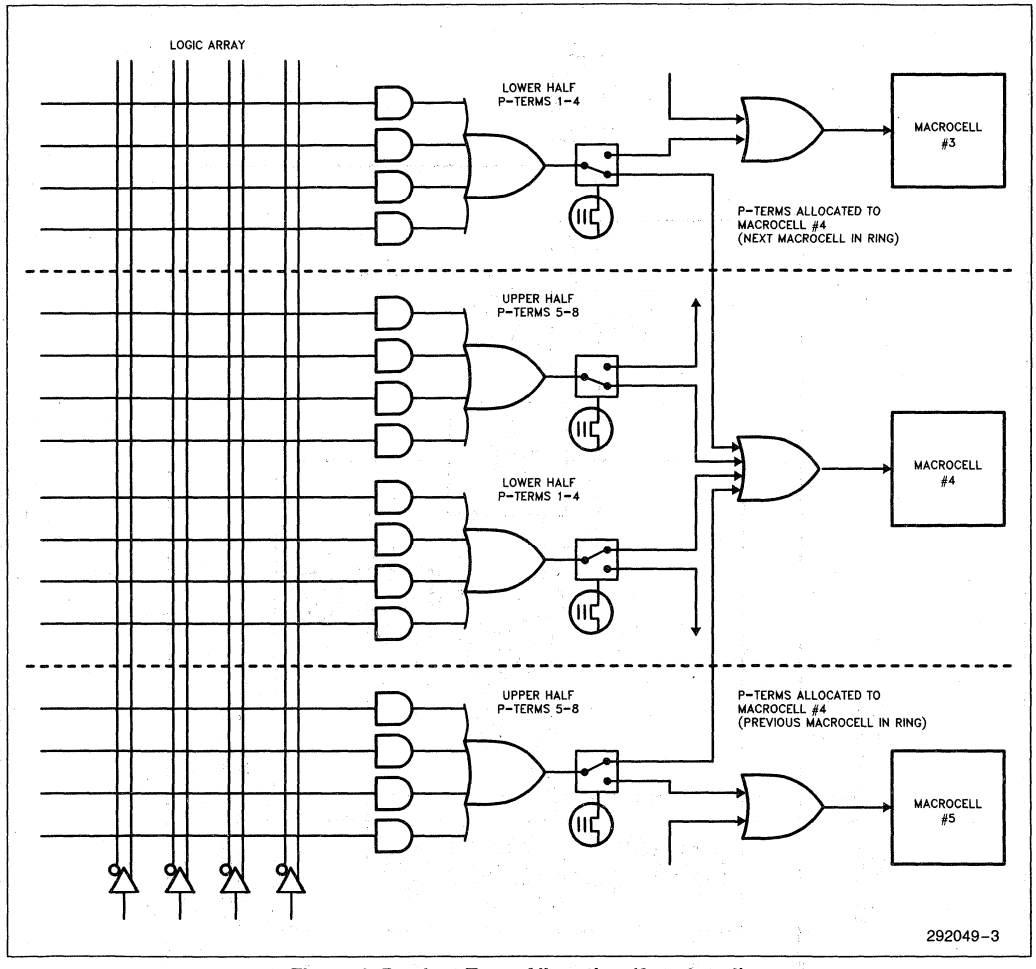


Figure 3. Product Term Allocation (8 + 4 + 4)

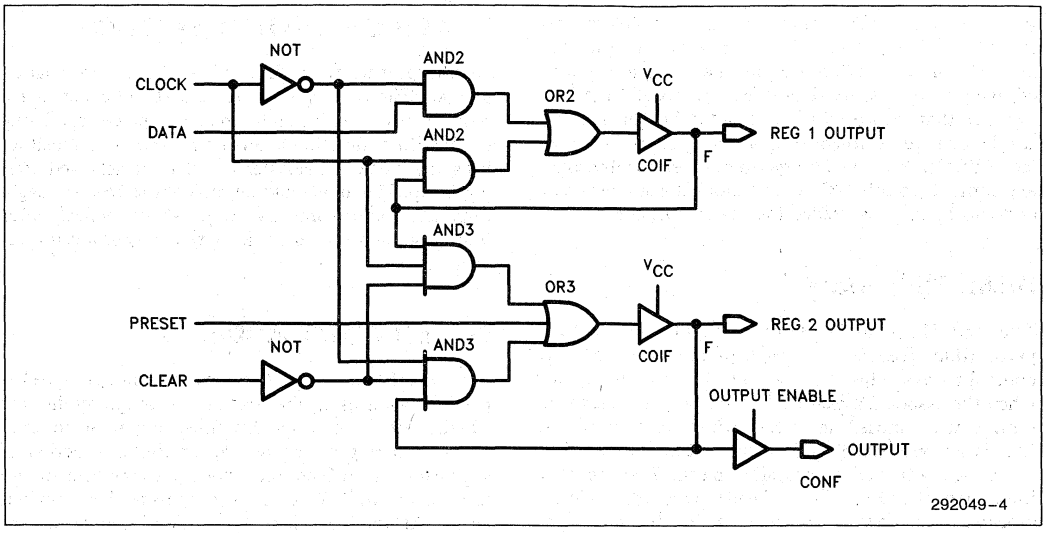


Figure 4. Implementation of D Flip-Flop with Added Preset Function Using Combinational Logic

of logic resources (p-terms) from areas they are not being utilized to other areas within the chip where they are needed. As shown in Figure 3, each macrocell has the potential to borrow 4 more p-terms to add to the 8 it already has from each of its adjacent macrocells. This increases the maximum number of p-terms per macrocell to 16. Thus, any macrocell within the 5AC312 has the potential to satisfy logic functions requiring between 0 and 16 p-terms.

P-terms can be allocated in a "shift register" mode within each of the two rings of the macrocell; however, allocation of p-terms between rings is not possible. See Table 1 for a listing of adjacent macrocells within p-term allocation rings.

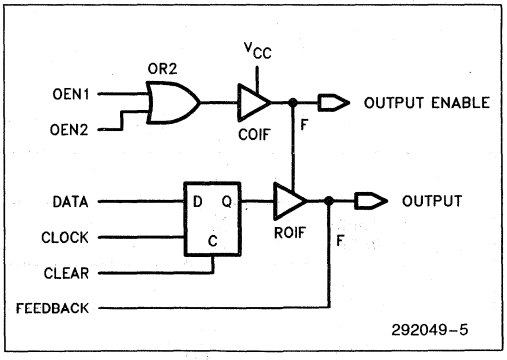


Figure 5. Implementation of 2 P-Term OE Control Signal

Table 1. 5AC312 Product Term Allocation Rings

Ring 1			Ring 2		
Current Macrocell	Next Macrocell	Previous Macrocell	Current Macrocell	Next Macrocell	Previous Macrocell
1	2	8	7	8	12
2	3	1	8	9	7
3	4	2	9	10	8
4	5	3	10	11	9
5	6	4	11	12	10
6	1	5	12	7	11

A given macrocell's output structure is still available for use when some or all of its p-terms are allocated away. If all of the p-terms of one macrocell are allocated away to its respective adjacent macrocells, the data input to that macrocell defaults to GND. This polarity can be changed through programming of the invert select EPROM bit. The I/O register as well as all secondary controls to this I/O control block are still available and can be used as needed for design purposes.

DUAL FEEDBACK

The 5AC312 contains separate input and feedback paths (dual feedback) on each of the macrocell I/O control blocks. This allows designs to utilize input pins when the associated macrocells have been assigned a no output with buried feedback primitive. Multiplexed I/O is accomplished by controlling the output buffer associated with each macrocell using the 2 p-terms that implement the OE function. Registered outputs may be clocked from the synchronous CLK/INP1 pin or asynchronously clocked by the 2 p-terms available for ASYNCH_CLK.

POWER ON CHARACTERISTICS

Another feature of the 5AC312 is its power-on characteristics. The I/O registers of the 5AC312 experience a reset to their inactive state upon Vcc power-up. Using the PRESET function available for each macrocell allows any particular register preset to be achieved after power-up. The inputs and outputs of the 5AC312 begin responding approximately 10 μs (6 μs typical) after Vcc power-up or after a power-loss/power-up sequence.

POWER DOWN MODE

A trade-off between power consumption and speed is possible when using the 5AC312 by programming the "Turbo Bit". Left unprogrammed and with no transition occurring at the device inputs for a period of approximately 100 ns, the device powers-down the internal array while holding the outputs at their previous levels. At the next input transition occurrence, the 5AC312 powers-up the array and reacts to the change in input conditions. If the "Turbo Bit" is programmed, the power-down circuitry is disabled and the device will not power-down even if there are no more transitions. The array power-up sequence requires an additional 20 ns of propagation delay. Power supply current during power-down is no more than 120 μA. See Figure 6.

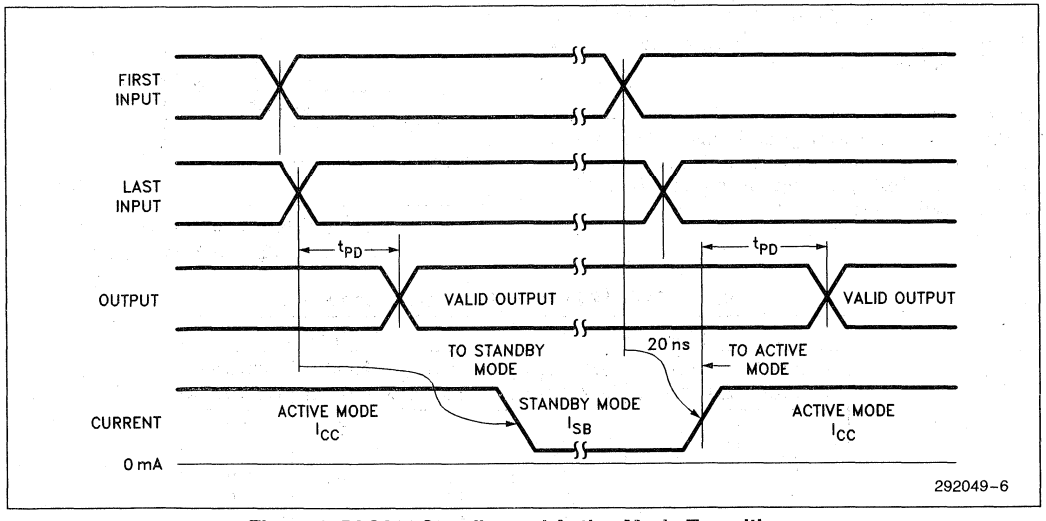


Figure 6. 5AC312 Standby and Active Mode Transitions

EXAMPLE

An example application for the 5AC312 can be shown by replacing a PAL* 20R6 and a 374 D type flip-flop in a design due to a power constraint. The same implementation can be achieved consuming less power using one 5AC312 EPLD. Compare Figures 7 and 8. Straight jumpers can be substituted in the PC board where the 374 sits, and since the clock signal is already available on the PAL socket, it can be internally routed to clock the input registers of the 5AC312. The 5AC312 can then be programmed to match the existing pin assignments and therefore require no PC board re-layout. The internal circuitry of the 5AC312 allows the EPLD to act as both a D type flip-flop and a PAL.

SUMMARY

The 5AC312 EPLD, which uses advanced CHMOS EPROM cells as logic control elements instead of polysilicon fuses, represents an innovative device to help overcome the primary limitations of standard PLDs. With its advanced features, proprietary architecture and macrocell structure, the 5AC312 is capable of implementing high performance logic functions more effectively than was previously possible. The p-term allocation scheme is a unique feature, increasing the efficiency of the device immensely. The PRESET signal and 2 p-term control lines are also features giving the 5AC312 added efficiency in many designs.

These same architectural features have been included in the 5AC324 EPLD, making that device ideal for even higher integration applications. Refer to the 5AC324 Data Sheet for details on that device.

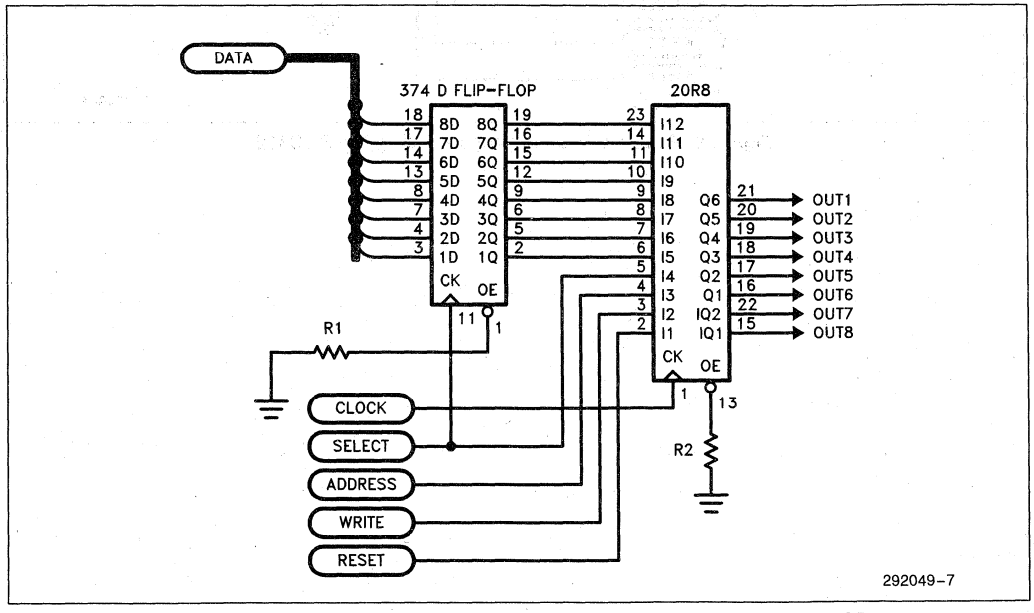


Figure 7. Original Implementation Using a 374 D Flip-Flop and A PAL2OR6

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

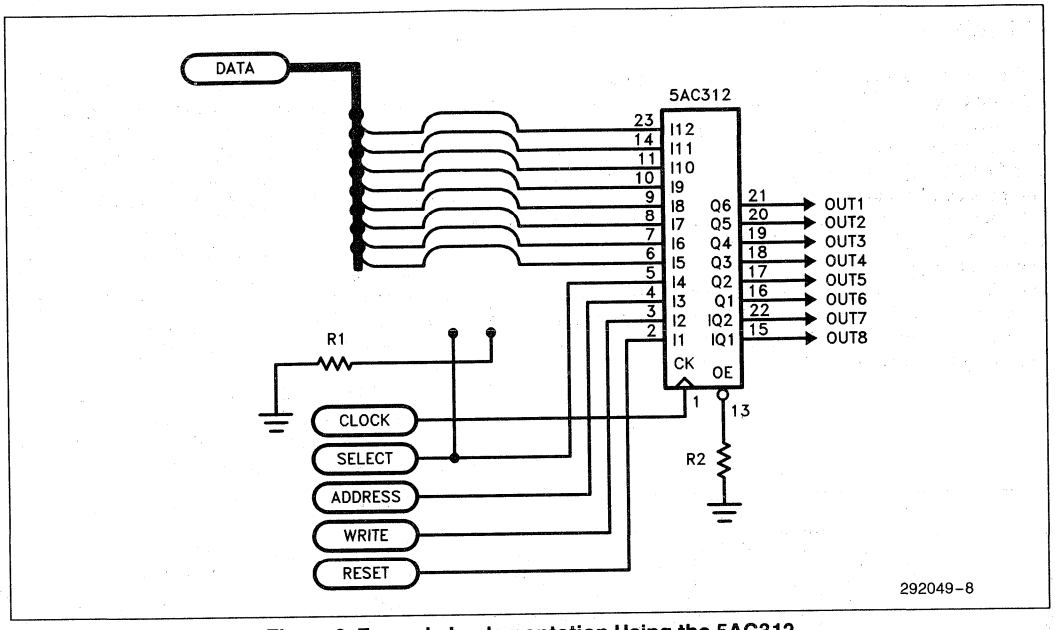


Figure 8. Example Implementation Using the 5AC312

IDEAS FOR DESIGN

A MICRO CHANNEL SLAVE-ADAPTER LINK

TODD K. KOELLING
Intel Corp., 1900 Prairie City Rd., Folsom, CA 95630; (916) 351-5788.

Many times neither of the two available implementations for executing a PS/2 Micro Channel slave adapter are optimum solutions. Precustomized PS/2 Micro Channel chips may offer more functionality than necessary for many slave-adapter designs. A standard PAL-and-TTL solution, however, can require too much of the valuable PS/2 adapter-board space.

For the Micro Channel slave-adapter interfaces that are too small for full-custom design, but too large for a PAL implementation, the 5AC324 offers a third solution. In addition to saving board space, the 5AC324 saves power. With a 50-mA I_{CC}, the CMOS part consumes much less power than its bipolar (or CMOS) PAL substitutes. A 40-pin, 24-macrocell programmable logic device (EPLD), the 5AC324 supplies high integration and includes input latches. Furthermore, designers can customize it for each interface design.

For example, consider an interface for an 8-bit I/O slave *(see the figure)*. The 5AC324's transparent input latches makes latching incoming address and bus-cycle information convenient and ensures valid data throughout the command cycle. Decoding the latched data generates the I/O and transceiver control outputs, ($\overline{IORD}$ and $\overline{IOWR}$ and $\overline{TOE}$). The inputs latch on the Address Decode Latch ($\overline{ADL}$) signal; the Command ($\overline{CMD}$) signal gates the outputs.

For read or write cycles, the 5AC324 EPLD first establishes the transceiver direction with the Transceiver Send/Receive (TS/$\overline{R}$) signal. Then, the 5AC324 enables the transceiver ($\overline{TOE}$) and generates the read/write strobe ($\overline{IORD}$ or $\overline{IOWR}$) when the command cycle ($\overline{CMD}$) becomes active. The latched Micro Channel bus-cycle information signals needed to generate these controls are the memory-I/O cycle (M/$\overline{IO}$), Bus Status ($\overline{S1}$, $\overline{S0}$), Card-Setup ($\overline{CDSETUP}$), and Board Select Decode (BDSEL) signals.

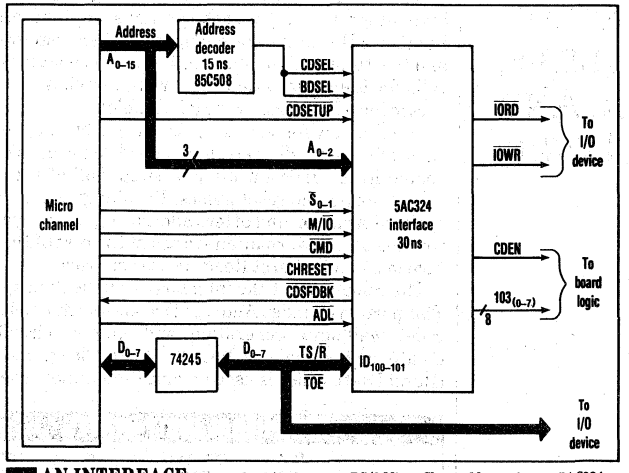
Since the $\overline{CDSFDBK}$ signal must be generated off the unlatched address decode, a second address decode (CDSEL) feeds into the EPLD. The CDSEL signal remains unlatched while BDSEL is latched for the control strobes. The address decoding could be performed inside the 5AC324. However, because that function requires many pins and little logic, an external 8C508 EPLD gives more efficient address decoding.

Besides generating the control signals, the 5AC324 also houses a significant amount of programmable option-select (POS) logic. The 5AC324 implements the POS IDs (POS 100 and 101), Card-Enable register (POS 102, bit 0), and option-select databyte 2 (POS 103). All POS data transfers through a multiplexed bus configuration on data lines 0 through 7 (D_{0-7}). The user programs the ID bytes to his unique card ID value.

The Card-Enable register establishes the required card-enable function, and the POS 103 register is available for user-defined configuration data. The Micro Channel-bus input signals that determine the POS cycle are the Card Setup ($\overline{CDSETUP}$), Lower Address lines (A2, A1, A0), Memory or I/O Cycle (M/$\overline{IO}$) signal, and Bus Status ($\overline{S1}$, $\overline{SO}$) signals.

To meet Micro Channel bus-driving requirements, the circuit uses an external 74245 bus data transceiver. This arrangement isolates the local data bus from the Micro Channel data bus. With a 15-ns address-decoding device, the 30-ns T_{PD} of the 5AC324 easily meets the timing requirements for the $\overline{CDSFDBK}$ and POS signals. Because timing requirements are fairly tight, an extended cycle would need faster circuitry outside the 5AC324.

A designer can easily modify this 8-bit interface into a 16-bit interface by adding an additional transceiver, Card Data Size 16 ($\overline{CDDS16}$), and controls for it. By utilizing the unused outputs and decreasing the POS logic as needed, the design could be expanded to control memory transactions. □



AN INTERFACE for an 8-bit I/O slave to a PS/2 Micro Channel bus using an 5AC324 EPLD can save board space and power.

Reprinted from **Electronic Design**, April 13, 1989, A Penton Publication.

DESIGN APPLICATIONS

SIMPLIFY A RISC EMBEDDED-CONTROLLER INTERFACE USING A PLD

A PLD IMPLEMENTS THE BURST LOGIC, TIMING CONTROL, TIMING ANALYSIS, LOGIC DESIGN, AND PROGRAMMABLE LOGIC.

THOM BOWNS
Intel Corp., 1900 Prairie City Rd., Folsom, CA 95630;
(916) 351-8080.

Reprinted from **Electronic Design**, January 25, 1990, A Penton Publication.

Designers are turning to reduced-instruction-set computer (RISC) processors, such as the 80960KB, for embedded-control systems because they deliver the necessary speed and streamlined operation. Design problems arise, however, when RISC processors are used. For example, RISC processors have more complex interface needs than do slower, conventional embedded controllers.

With the 80960KB, designers can take advantage of the 85C960 programmable-logic device, which implements the difficult portions of the interface and easily configures the programmable logic. The difficult portions that are handled by the application-specific PLD include the burst logic and timing control with all of their state tables, timing analysis, and logic design. The programmable logic implements flexible address decode and complete wait-state coverage.

The 80960KB's burst-mode bus and high-performance RISC architecture make system designs efficient in terms of size and space. But linking the chip with peripheral devices, such as memory or I/O devices, can require a fair amount of circuitry. Some portions of the interface vary little from one application to another, while other parts are customized according to the application. Designing this type of interface can be difficult and time consuming.

The basic 80960KB interface circuitry is composed of several discrete sections, including address latching, address decode, data transceivers, burst bus-control logic, wait-state generation and Clock/Reset synchronization *(Fig. 1)*. The application-independent sections—address latching, data buffering, Clock/Reset synchronization, and burst bus-control logic—require high-performance, high-integration logic. These portions won't necessarily be affected by peripheral changes or other customization and fine tuning of the system. The address-latch portion calls for four byte-wide flow-through latches and an inverter for $\overline{\text{ALE}}$ (Address Latch Enable).

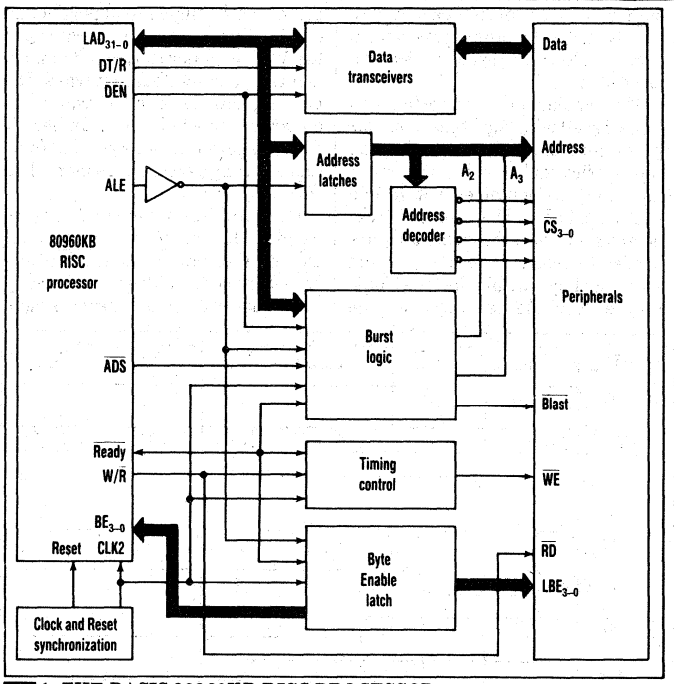
The circuitry for the Clock/Reset synchronization consists of several discrete high-speed logic devices and should have sufficient drive to supply all of the necessary clocking and reset pulses. Burst logic decodes bus-cycle information and offers addressing, burst cycle status, and other information –to assist the rest of the interface in handling burst Read/Write accesses.

The remainder of the interface is application-specific. This portion includes Address Decode (or Chip Select Decode), wait-state generation, and optional circuitry to create such signals as $\overline{\text{WCLK}}$ (Write Clock) and $\overline{\text{Blast}}$ (Burst Last), which is used by burst-mode memories. The

PERIPHERAL WAIT STATES

	First read	Subsequent reads	First write	Subsequent writes
Flash	2	2	5	5
Burst EPROM	1	0	–	–
SRAM	1	1	1	1
I/O	3	3	4	4

PLD-BASED RISC INTERFACE



1. THE BASIC 80960KB RISC-PROCESSOR interface consists of application-independent and application-dependent sections. Portions that remain the same regardless of the application include address latching, data buffering, Clock/Reset synchronization, and burst bus-control logic.

address decoder, which is programmed to fit designers' memory maps, supplies latched Chip Selects to the various peripheral circuits. The wait-state generator reads the Chip Selects to determine which wait state count value to use, and then looks for a "go" signal from the bus state tracker to begin counting out wait states. $\overline{\text{Blast}}$ and $\overline{\text{WCLK}}$ are generated by way of state machines that work closely with the burst-control logic.

TRADE-OFFS AND PITFALLS

Designers must balance several trade-offs affecting the interface circuitry that ties the 80960KB RISC processor to its peripheral devices. The address decoder, for example, must be flexible enough to fit designers' memory maps, and it must be fast enough to select the peripherals without adding to the peripheral access time. If the decoder is placed after the address latches, the timing path would include a clock to $\overline{\text{ALE}}$ (T_{CO}), latch delay, decode delay, peripheral-access time, data-transceiver delay, and setup time to the sampling edge in the data state (T_{CO}) *(Fig. 2a)*. The cumulative delay of T_{ACC} equals four CLK2 periods minus T_{CO} and T_{SU}. In a 20-MHz system with zero wait states, this would mean a period of 100 ns minus 23 ns, leaving no more than 77 ns for latch, decode, peripheral access, and transceiver delays.

The wait-state generator also involves some trade-offs: compactness and complexity. The wait-state generator should cover all of the possible wait-state requirements for the various peripherals, which include differentiating between read and write accesses, and first and subsequent accesses within a burst transfer. The amount of logic needed to supply comprehensive wait-state coverage depends on the application. A single-PLD counter alone can handle only a few wait-state count values. Another approach would be to have individual PLD wait-state counters associated with each peripheral. Each wait-state counter's output would feed a main ready generator.

Nevertheless, only three clock edges are encountered from address time to the end of a data or wait state. The circuitry must decode peripheral selects, load in the correct wait-state count value, count down, and return a ready/not ready indication before the setup to the sampling clock edge *(Fig. 2b)*.

Generation of the $\overline{\text{Blast}}$ signal for burst memories is rather tricky because the signal must occur during the data state of the last transfer. The circuitry controlling this signal must read the burst-progress information from the burst-logic section, and then assert $\overline{\text{Blast}}$ during the data state. The only way to determine whether the current state will be a data state or a wait state is to look ahead at the ready generator and anticipate its assertion by one CLK2 cycle *(Fig. 2c)*.

$\overline{\text{WCLK}}$ must be qualified with the Latched Byte Enables to write to individual bytes of static RAM (SRAM). Because this will delay $\overline{\text{WCLK}}$ to some degree, it's necessary to delay A_2 and A_3 relative to $\overline{\text{WCLK}}$ to achieve a correct timing relationship between those signals *(Fig. 2d)*.

The heart of the interface is the burst-logic section. It consists of several counters and state machines, which include the burst-size counter,

PLD-BASED RISC INTERFACE

the address counter (A_2 and A_3), and the $\overline{\text{Blast}}$ generator. The burst logic must sense the beginning of a burst access, load the burst-size down-counter from local address lines LAD_0 and LAD_1, and store the initial low-order address values that were read from LAD_2 and LAD_3 in the address counter.

As each transfer is completed—signaled by the assertion of $\overline{\text{RDY}}$—the burst-logic circuitry increments the address counter, decrements the burst-size counter, and determines whether additional transfers remain in the burst access. If there are more burst transfers in the current burst cycle, the burst logic signals the wait-state generator and the other state machines to proceed with another transfer.

To further complicate matters, these counters and state machines must be made codependent on each other and the rest of the interface. Designers must carefully plan each element in the burst logic, keeping in mind the states and timings of signals from the other portions of the interface.

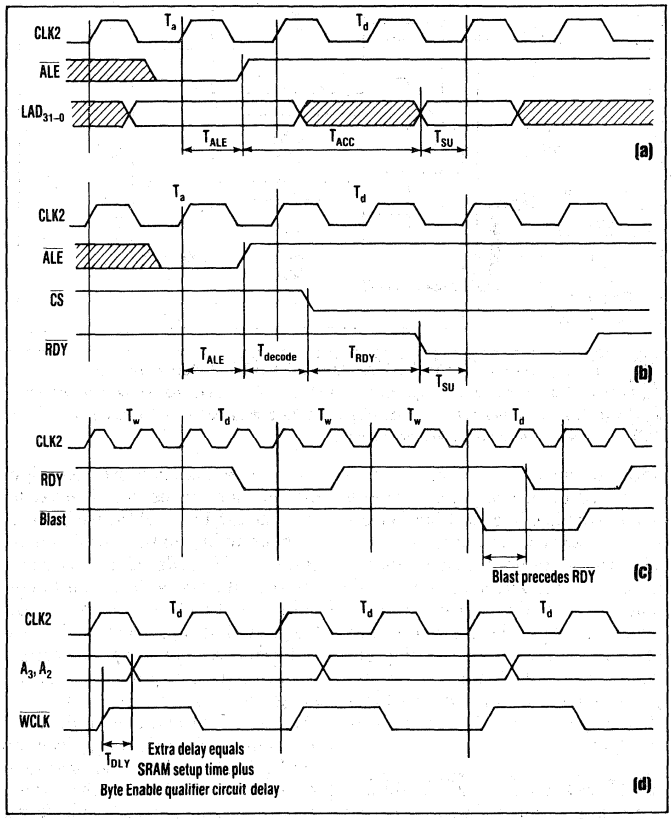
PROGRAMMABLE LOGIC

Interface circuitry is typically implemented in fast discrete TTL devices and bipolar programmable-logic devices. When using TTL and programmable logic, most designers break the circuit into discrete gates whenever possible. The remainder of the circuit is described in terms of registered functions or state machines.

Most of the application-independent areas of the interface can be handled by high-speed TTL. The address latches and data transceivers would most likely be constructed of fast TTL devices, such as the 74F373 and 74F245, respectively, along with a 74F00 that would be used to invert $\overline{\text{ALE}}$. A 74F138 could generate Chip Selects by decoding the latched high-order addresses. Several fast logic gates and registers could generate the full-speed and half-speed clock signals to synchronize Reset, and to supply these and other synchronization signals to the rest of the system as needed.

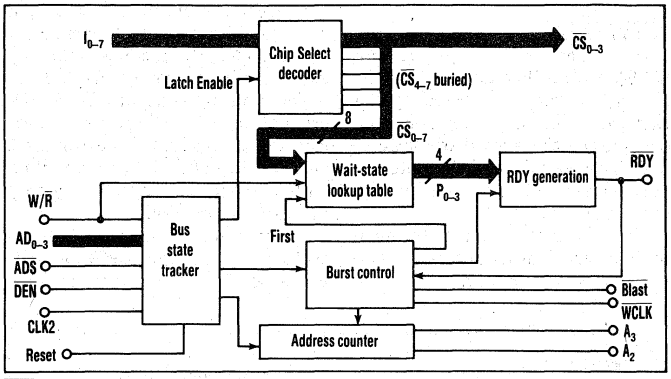
The remainder of the interface can be designed with high-speed, bipolar programmable logic. Depending on the application's needs, the burst logic may be contained within one high-speed PLD. The number of registers and combinatorial outputs needed to produce the counters and state machines within the burst logic may exceed the resources of one PLD. The timing-control portion, including the wait-state generator and the $\overline{\text{WCLK}}$ generator, can be implemented in many different ways. Depending on the level of wait-state coverage needed, the number of PLDs required to perform this function can range from one to more than seven devices.

One middle-of-the-road way to handle the wait-state generator would be to have a main $\overline{\text{RDY}}$- and $\overline{\text{WCLK}}$-generator PLD that's fed by individual PLDs, one for each periph-



2. THE TIMING RELATIONSHIPS between the interface signals are critical (a, b, c, d). For example, the interface circuit must decode peripheral selects, load in the correct wait-state count value, count down, and return a ready/not ready indication before the setup to the sampling clock edge.

PLD-BASED RISC INTERFACE



3. BOTH HARD LOGIC and programmable resources are offered in one package with Intel's 85C960 PLD. It's an application-specific part designed to perform many of the 80960KB processor's interface functions.

eral. As a peripheral Chip Select is generated, a corresponding PLD is activated and sends an appropriate wait-state count value to the main $\overline{RDY}$ generator programmable-array logic (PAL). That PAL then counts down from the value it received and asserts $\overline{RDY}$ at the end of the count. Each individual PLD could then distinguish between first and subsequent accesses within a burst, and between read and write accesses. With this information, the programmable-logic device could furnish a proper wait-state value.

The easiest way to create the burst logic is to determine the various elements—the burst-size counter and bus-state tracker, the address counter, and the burst-control logic—and describe each element in terms of state diagrams. The processor-bus states must be understood to properly design the bus-state tracker and relate it to the other state machines. The timing-control portion consists of individual programmable-logic devices that handle the various wait-state counts, and a main $\overline{RDY}$ down-counter PLD, which also creates $\overline{WCLK}$.

This 80960KB-based design requires the $\overline{WCLK}$ generator to be described as a state machine, while the $\overline{RDY}$ generator is just a pre-loadable down-counter with additional start, stop, and reset logic. By fully describing the burst logic and timing control as a set of linked state machines, designers can determine the amount of PLD resources needed. The state-machine descriptions are translated into PLD source language, either directly into state-machine or other high-level format, or as minimized Boolean equations.

PARTITIONING

Designers then partition the design to fit into the available PLD resources and create simulation vectors to test the design's individual elements. The next step is to compile and simulate the resulting source files, which are programmed into the PLDs after debugging.

One major disadvantage of this solution is the address decode's inflexibility. The Chip Selects generated by the hard-logic device don't allow for easy restructuring of the memory map when additional memory is added or existing mapping is swapped. The inflexibility, combined with the tedium of designing the counters and state machines, partitioning all of that logic into the available PLD resources, and trying to end up with a compact, low-power solution, creates lots of extra time and effort that could be spent improving the system in other areas.

Intel's 85C960 PLD is an advanced, CMOS, application-specific PLD that performs the major interface functions for 80960KA/KB-based systems. This PLD doesn't perform the entire 80960KA/KB interface, but it contains select decode, timing control, and burst logic in one 28-pin package. The 85C960 PLD was designed to deliver high-performance control for burst cycles without sacrificing configurability. Application-specific PLDs are the only programmable devices that offer high-performance, high-integration logic with a satisfactory degree of flexibility.

The chip contains an eight-input, eight-output (four external) address decoder *(Fig. 3)*. Each output is independently configurable to respond to any address condition and is latched by an internally generated Latch Enable. In addition to having the flexibility of complete programmability, the address decoder generates the Chip Selects as fast as 10 ns after it receives valid addresses.

The PLD also has a programmable wait-state generator with a comprehensive–lookup table. There are four wait-state values associated with each Chip Select. When a given select is asserted, one of the four values associated with that Chip Select is loaded into the wait-state counter. The value depends on whether the current access is the first or second through fourth in a burst, and whether it's a Read or Write. This supplies the complexity to cover any wait-state requirement for each peripheral addressed by the PLD without the need for vast amounts of individual logic devices.

The burst logic is also on the chip. This incorporates the bus-state tracker, address cycling, and other timing-control logic, such as $\overline{WCLK}$, and $\overline{Blast}$. During a burst access, the bus-state tracker generates an inter-

PLD-BASED RISC INTERFACE

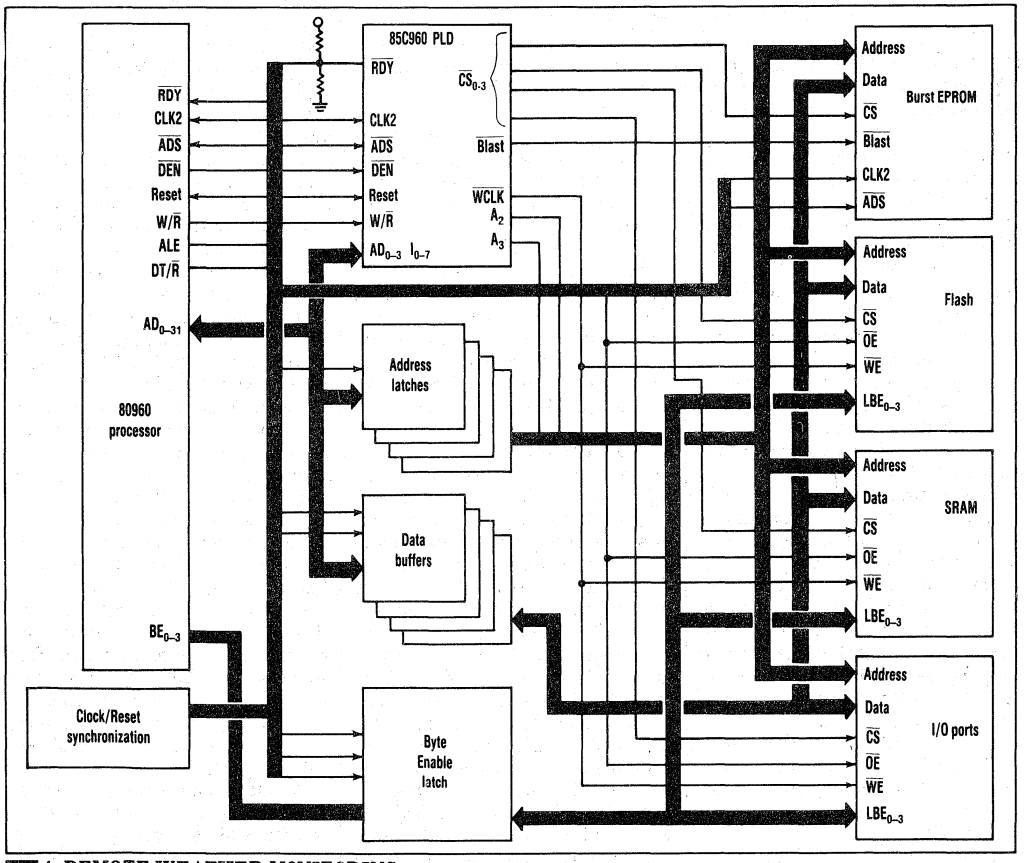
nal Address Latch Enable for the Chip Select decoder, and loads the wait-state generator with the appropriate count value. The tracker reads the low address inputs for burst size and starting address, then places A_2 and A_3 out on the address bus. During the burst, it keeps track of how many accesses remain in the burst, cycles A_2 and A_3, strobes $\overline{WCLK}$ appropriately, recycles the wait-state generator, and asserts $\overline{Blast}$ while the last access is occurring.

Burst Logic

One great advantage of the 85C960 PLD is that the burst logic is already in place and tested by the factory. The hassles of designing the discrete solution with all of the state machines, equations, timing interdependencies, and logic partitioning are eliminated.

The PLD's $\overline{RDY}$ pin is an open-drain driver. During accesses when the PLD isn't generating a Chip Select Decode, it will read the $\overline{RDY}$ pin as an input and cycle the burst logic appropriately to produce valid A_2, A_3, and $\overline{WCLK}$ timing. A_2 and A_3, which have high-output CMOS drivers that can drive a large memory array, are timed with $\overline{WCLK}$ to make it possible for $\overline{WCLK}$ to be qualified with Latched Byte Enables by way of external circuitry.

An example of an embedded-control system using the 80960KB RISC processor is a control system for a



4. REMOTE WEATHER MONITORING is performed by this embedded system built with the 80960KB processor, memory, and I/O devices. A PLD implements the interface between the processor and other elements.

PLD-BASED RISC INTERFACE

remote weather monitoring unit *(Fig. 4)*. The unit might be placed in far-flung locations to monitor weather conditions, perform high-speed calculations, and then quickly return digested data to the main weather station.

The weather monitoring unit features a video-display-terminal interface, a serial microwave-transmitter link, and sensor interfaces. This type of system could operate continually in various climate conditions, and perhaps for long periods on backup power. As a result, the system would need highly reliable, low-power components. Because the unit is in a remote site, it would be checked for routine maintenance perhaps only twice a year. It should have some way of receiving new instruction code over the microwave link and storing the code in nonvolatile memory. If a hardware failure is detected, the single embedded-controller board could be removed and replaced in minutes.

The unit is built around the 80960KB embedded-control system with memory and I/O devices. Burst EPROM takes full advantage of the speed and width of the processor's burst bus. Flash memory is used for nonvolatile data storage and for updatable-code space. A certain amount of SRAM is needed by the processor for stack and scratch-pad memory. And with the I/O ports, the embedded controller can communicate with the various weather sensors, the video-terminal link, and the microwave transceiver.

Designing The Interface

The processor is interfaced to the four peripheral subsystems by way of the PLD and the other circuitry. The local address/data bus runs through a set of TTL transceivers to the system data bus, and through TTL latches to feed the system address bus. Byte Enable signals are latched in a high-speed PLD and routed to the peripherals that need them. The processor supplies the remainder of the interface, which includes the address decode, wait-state generation, burst control, $\overline{WCLK}$, and $\overline{Blast}$ generation.

The local (unlatched) address/data lines, LAD_{31-24} and LAD_{3-0}, are brought into the PLD along with the processor's bus-control signals $\overline{ADS}$, $\overline{DEN}$, CLK2 (high-speed clock), Reset, and W/$\overline{R}$. $\overline{RDY}$ is terminated with a pull-up/pull-down network, and feeds the processor's $\overline{Ready}$ input as well as the Byte Enable latch. To facilitate burst control, non-burst peripherals receive A_2 and A_3 from the PLD rather than from the address latches. W/$\overline{R}$ from the processor ties directly to the Output Enable inputs for all of the peripherals, and $\overline{WCLK}$ from the PLD feeds the Write Enable lines for the peripherals that require a Write clock. $\overline{Blast}$ runs from the PLD to the burst-EPROM bank, along with $\overline{ADS}$ and CLK2.

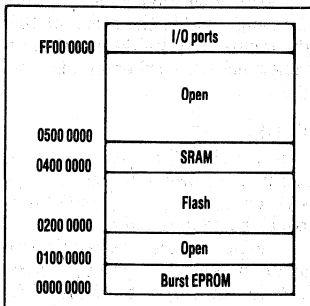
According to the system-memory map, the burst EPROM is located at the bottom of memory for boot up *(Fig. 5)*. Flash memory is located higher up, followed by SRAM. To allow for expansion, a large open space follows SRAM. The I/O ports are located at the very top locations of memory.

The wait-state values for all of the peripherals vary under the different burst-transfer conditions *(see the table)*. Note that burst EPROM needs one wait state for initial reads, but needs no wait states for successive reads. Flash memory, with a 135-ns access time, requires two wait states for either reads or writes. Writing to the flash memories is accomplished by performing a command-data algorithm of writes and reads.

Design Considerations

There are several important items to consider when designing with the 85C960 PLD. Standard CMOS-design practices should be observed, such as a healthy bypass capacitor of at least 0.2 μF placed as close as possible to the V_{CC} pin. Any unused inputs should be tied low to avoid high-power consumption. During normal operation, V_{CC} must be ramped up to full voltage before any inputs begin to toggle, otherwise the device may enter an unknown state.

The timing of Reset versus CLK2 is critical. The 85C960 PLD relies on an internally generated clock signal which should be in phase with the bus clock. The first rising edge of CLK2 after Reset falls is considered by the PLD to be the "a" edge—the



$EPROMCS = !A_{31} * !A_{30} * !A_{29} * !A_{28} * !A_{27} * !A_{26} * !A_{25} * !A_{24}$

$FLASHCS = !A_{31} * !A_{30} * !A_{29} * !A_{28} * !A_{27} * !A_{26} * A_{25}$

$SRAMCS = !A_{31} * !A_{30} * !A_{29} * !A_{28} * !A_{27} * A_{26} * !A_{25} * !A_{24}$

$IOCS = A_{31} * A_{30} * A_{29} * A_{28} * A_{27} * A_{26} * A_{25} * A_{24}$

5. THE SYSTEM memory map illustrates that the burst EPROM is located at the bottom of the memory for system boot up. The equations are derived from the address locations shown.

PLD will time all of the inputs and outputs with this reference. Reset timing is consistent with the timing specification for the 80960KB processor.

Though the 85C960 PLD decodes eight Chip Selects, only four of them are routed externally. This might be troublesome for systems which have five or more peripherals. The easiest way to work around this is through the use of an additional, external ad-

PLD-BASED RISC INTERFACE

dress decoder, such as the Intel 85C508 programmable-address decoder. The external decoder could be programmed to supply Chip Selects in the ranges that shadow those given by one or more of the PLD's buried Chip Selects. This would make it possible for the PLD to continue to supply bus control for those peripherals as well. System designers should take note that the PLD Chip Selects are decoded with one product term. This affects the mapping of peripherals such that while any address range within the scope of the eight inputs may be decoded, very complex mapping schemes may be impossible.

PULL-UP

The I/O pin $\overline{RDY}$ must be terminated in close proximity to the processor using the pull-up/pull-down network listed in the data sheet, unless loading on this net requires a heavier pull-up. The pull-up determines the ramp-up rate of the $\overline{RDY}$ net. If the pull-up is too light or a capacitive load is too heavy on the $\overline{RDY}$ net, $\overline{RDY}$ may rise too slowly and not meet the processor specification on $\overline{RDY}$ setup. Any other wait-state generators driving the $\overline{RDY}$ net must do so by way of the open-drain drivers to avoid contention with the PLD.

Although the burst logic is cycled appropriately when the 85C960 PLD isn't generating the Chip Select or $\overline{RDY}$ signal, $\overline{Blast}$ doesn't get produced in these circumstances. This is due to the timing relationship between $\overline{Blast}$ and the sampled $\overline{RDY}$ input *(Fig. 2c, again)*. $\overline{Blast}$ must be asserted at the start of the last data state, and $\overline{RDY}$ isn't necessarily valid until close to the end of the last data state. However, because the PLD will produce $\overline{Blast}$ during accesses for which it decodes Chip Selects, designers merely need to ensure that the PLD decodes the Chip Selects for the burst memory.

The only portions of the 85C960 PLD that need configuration are the address decoder and the wait-state lookup table. The address decoder reads the eight address inputs, latches them, and decodes the address range according to the equation programmed in the product term of each individual Chip Select output. The Chip Selects then help determine lookup table will be loaded into the $\overline{RDY}$ counter.

The Chip Select decode equations must be created first. In the example design, the Chip Select equations can be translated from the information given in the memory map *(Fig. 5, again)*. Because eight address lines are brought into the Chip Select decoder, using the most significant eight address signals from the 80960KB processor allows for the widest range decode.

DERIVING EQUATIONS

The equations are derived by taking the product of the true, or the complement, of the address lines that must decode a given block of memory. For instance, burst EPROM begins at address 00000000h. The most significant byte is 00, meaning that the upper addresses—LAD_{31-24}—would be 00000000b. This produces the equation labeled EPROMCS. The memory block for flash memory requires twice the decode range than the burst EPROM block. Because the PLD is limited to one product term per output, the flash-memory block begins at a location following an open space after the burst-EPROM block. The equation decoding flash memory (FLASHCS) disregards the lowest of the incoming address lines.

The wait states for each peripheral are derived by determining the access-time requirements for each device and comparing that against the allowable time during each transfer state. Suppose, for example, that the allowable access time (including the decode, latch, and transceiver delays) for zero wait states is 55 ns. If the access time of the SRAM used in this design is 60 ns, at least one wait state would be required. An additional 50 ns are gained by the one wait state, making it possible for the use of up to 105-ns SRAMs.

The burst EPROM needs one wait state for address setup on the initial read access. However, subsequent reads during a burst transfer aren't required to setup addresses to the device because that's taken care of internally. Therefore, there are zero wait states in the subsequent burst transfers. The numbers for the rest of the peripherals are determined similarly.□

Thom Bowns, technical support specialist for the programmable logic group at Intel Corp., has an AS in digital and microwave electronics from American River College, Carmichael, Calif.

Programmable AND/ Allocatable OR Based EPLD Addresses the Needs of Complex Combinational and Sequential Designs

by Todd K. Koelling
Applications Engineer

INTRODUCTION

Matching programmable logic applications with programmable logic devices has become a difficult task. Increasing demands for higher integration, higher performance and lower cost continue to drive system design engineers on to new technologies. The programmable logic industry has adeptly responded by supplying a wide variety of devices. At times, however, it is hard to differentiate these devices and to determine which makes the best solution for a particular application.

In a small way, this paper will attempt to differentiate devices and to determine which devices make the best solutions for groups of applications. This task will be accomplished by taking a general look at applications, the history of PLD arrays and a new device which solves several design problems.

APPLICATIONS

College textbooks [1] on digital design teach that fundamentally there are only two types of applications: combinational and sequential. A combinational circuit generates outputs based on the immediate status of a group of inputs. A sequential circuit uses some mechanism to store data before generating the next set of outputs.

Inside combinational and sequential circuits are two fundamental elements: gates and registers. Gates are the prime component of combinational circuits where the output is an immediate function of the input. Registers are the static storage element added in sequential circuits to latch and hold data until the next cycle.

Figure 1 displays gates and registers graphically. The coordinates measure registers along the x-axis and gates along the y-axis. In this space, any combinational or sequential application can be displayed.

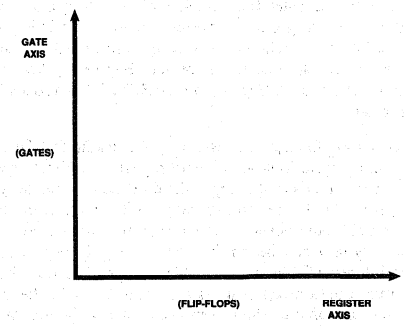


Figure 1. Gate/Register Coordinates

Common TTL functions are easily graphed. Figure 2 displays a comparator, storage register, shift register and counter. The comparator is a combinational circuit (purely gates) and hence lies along the gate axis. The storage register, on the other hand, is purely flip-flops and hence lies along the register axis. The shift register is primarily flip-flops—placing it close to the storage register—but it includes some gate logic, thus moving it up the gate axis. The counter is a good example of a function that lies somewhere in-between the two axes. The counter must store its current state, and thus leans heavily upon the registers, but it also uses a significant amount of gate logic to generate the next count state. The inclination toward the gate or register axis depends on the features the counter incorporates. Up and down count operation, clear and preload functions, and count enable/disable circuitry, all move the counter increasingly toward the gate axis. The magnitude of the counter (as with the other functions) depends

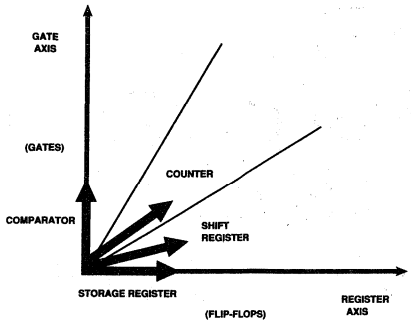


Figure 2. Common Functions on Coordinates

on the number of bit and features it includes. That is, a 16-bit counter is twice as large as an 8-bit counter which is twice as large as a 4-bit counter, provided the feature set remains the same.

Having examined functions, it is easy to examine complete designs. Each design can be decomposed into a group of function blocks. This is basically the undoing of the design process in which functions are grouped together to meet some high-level purpose. If all the functions for a design are added together, a vector for the complete design is formed (Figure 3). Now, rather than just a MAGNITUDE (size), a DIRECTION is also available for each design.

With designs defined and considered, it is useful to take a step back and once again look at the gate-register coordinate system. Any design that lands in the upper third of the coordinate system consists primarily of gates and thereby can be considered "highly combinational." Any design that lands in the lower third of the coordinate system can be considered as "highly registered" or "register intensive." Examples of this include data transfer and data storage applications. Any design that lands in the center third of the coordinate system represents a healthy mix of both gates and registers. This means it is probably a state machine or some sort of sequential application. Hence, the middle third region will be called the "state machine" region, though some state machines may land in the other two regions. The coordinate system with the three regions segmented and labelled is shown in Figure 4.

ARRAY ARCHITECTURES

Through the years, programmable logic devices have evolved by trying to meet the needs of combinational and sequential applications. This has been accomplished through higher integration, higher flexibility and higher performance and has resulted in the myriad of PLDs available today. Though the features have varied and expanded immensely, the core of the programmable logic device has remained virtually the same. It is this core—the implementation of the combinational logic array—which deserves a closer look.

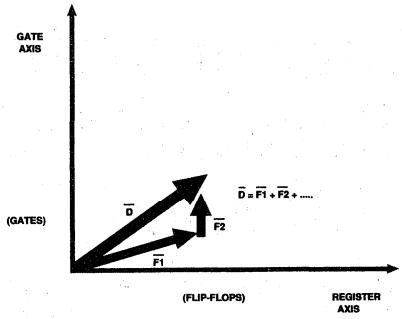


Figure 3. Design Vector

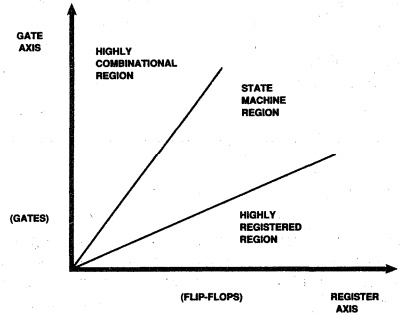


Figure 4. Application Regions

A Brief History of PLD Array Architectures

Programmable logic first appeared in 1975 with the introduction of the Field Programmable Logic Array (FPLA). With its programmable AND/programmable OR array, the FPLA was extraordinarily powerful for integrating combinational logic. Registers were later added to the FPLA outputs, creating the Field Programmable Logic Sequencer (FPLS). This device better attacked the needs of sequential applications.

In 1978, the FPLA was honed to a programmable AND/fixed OR array with the introduction of the PAL [2] device. Due to its ease of design and CAD tool support, the Boolean sum-of-products part quickly became the designer's choice. In addition, the PAL included registered versions with registered outputs and registered feedback. These two attributes made the devices ideal for state machines.

In essence, the last array architecture, fixed AND/programmable OR, has been around for many years in the form of the PROM and other PROM-based logic devices. It wasn't until 1984, however, that the SRAM-based LCA introduced the fixed AND/programmable OR array to the designer in an expedient form. Due to its large number of registers per device (122 or more), the LCA has provided an excellent programmable solution for register-intensive applications.

Mapping of Array Architectures

Returning to the application graph developed earlier, each architecture and device is displayed (Figure 5). The combinational power and flexibility of the programmable AND/programmable OR architecture places the PLA and PLS devices in the highly combinational region.

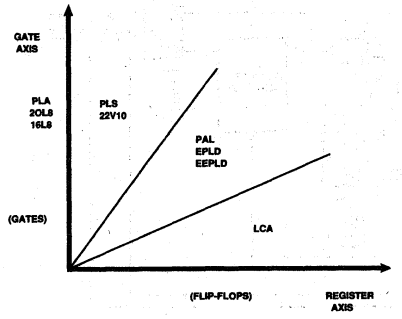


Figure 5. Device Application Areas

The registered output with registered feedback PALs are optimum for state machines while some PALs, such as the 22V10, are geared to provide a large amount of combinational logic per register. Thus, most of the registered PALs fit in the state machine region, while a few, like the 22V10, move up into the highly combinational region. (Logic PALs such as the 16L8 and 20L8 fit along the y-axis as they are purely gates.)

Architecturally, most EPLDs and EEPLDs have followed the track set by the PAL. They have added higher input and output flexibility, higher integration, and other worthwhile features, but the core of the architecture has remained the same—the programmable AND/fixed OR array. Thus the EPLD and EEPLD fit the same application regions as the PAL.

The highly registered region is best covered by the LCA where its high register count and fast toggle rates are well utilized.

Where will the next innovation be? The answer may not be in an innovation, but in a very practical advancement from Intel in 1988.

THE PROGRAMMABLE AND/ALLOCATABLE OR ARRAY

Via a novel product term allocation [3] scheme, the best of the PLA and the PAL have been married into a single device. In the programmable AND/allocatable OR array architecture, each macrocell output can have anywhere from 0 to 16 AND terms allocated to its OR gate in increments of four. In this manner it is very similar to the PLA with its OR input flexibility, yet it retains the Boolean sum-of-products architecture used in the PAL which offers easy design entry and optimized CAD tools for minimization and fitting. The net result is a novel approach for efficiently addressing the needs of complex combinational as well as sequential applications.

How P-Term Allocation Works

Figure 6 displays a macrocell of the 24-pin, 12 macrocell 5AC312. Each macrocell contains eight product terms grouped into two blocks of four. Each block has its own control switch that allows it to be retained by the macrocell or lent to a neighbor. Likewise, each macrocell can ignore or borrow a block from each of its neighbors. With two neighbors for each macrocell, each macrocell can have a total of 0, 4, 8, 12 or 16 p-terms allocated to its OR gate. This allows the device to shift resources toward those equations that require a high number of p-term inputs away from those that do not.

Combinational Example: Micro Channel [4] Decoder

Figure 7 shows the 5AC312 pinout for the micro channel decode logic on a PS/2 [4] Ethernet [5] adapter. Based on the address lines, micro channel bus cycle signals, and POS/command register inputs, the 5AC312 generates the card select feedback (CDSFDBK#) and card data size 16 (CDDS16#) return signals for the micro channel. It also generates an asynchronous board select signal (ABDSEL#) that feeds an on-board arbiter. As memory is shared between the PS/2 motherboard and the adapter 82586 LAN coprocessor, the equations for all three of these outputs become quite complex (Figure 8).

Even after processing the equations through the industry-standard Espresso [6] minimizer employed by the Intel Logic Optimizing Compiler (LOC), the nested equations in Figure 8 expand out into the minimized equations in Figure 9. The board select, card select feedback, and data size 16 signals are 15, 14 and 14 product terms, respectively. This is not a problem for the 5AC312, however, as the LOC software automatically allocates the device's resources to four, four p-term blocks for each signal. The three equations use a total of 12 out of the 24 four p-term blocks available in the device—50% of its combinational capacity.

Though not used in this application, since the CDSFDBK#, CDDS16#, and ABDSEL# are all driven off unlatched address decodes, another feature which makes the 5AC312 attractive for address decoding and bus interfacing is its latched input capability. On the IBM PC/AT [7] bus, for example, the upper address lines (LA17–23), may need to be latched with the active edge of the bus address latch enable (BALE). Elsewhere in the micro channel interface, latching the address and status signals may be useful as both will go invalid about halfway through the command (CMD#) cycle. The 5AC312 is capable of latching up to eight inputs, each of which can be enabled individually or by the global latch enable.

The decode design could be implemented in a PLA or a 22V10, but the 5AC312 offers more strength over the PLA and more flexibility over the 22V10. Even with 32 and 48 p-terms feeding the programmable OR gates, most 24-pin PLAs would have trouble fitting these equations. With its fixed allocation architecture of 8, 10, 12, 14 and 16 p-terms, the 22V10 offers p-term resources comparable to those of the 5AC312 and can handle the equations. With configurable p-term allocation, however, the 5AC312 offers a more flexible solution. In cases where latching needs to be done, the address latching must be done external to the 22V10 since it does not have the ability to directly latch inputs. This impacts both board space and performance in a negative fashion.

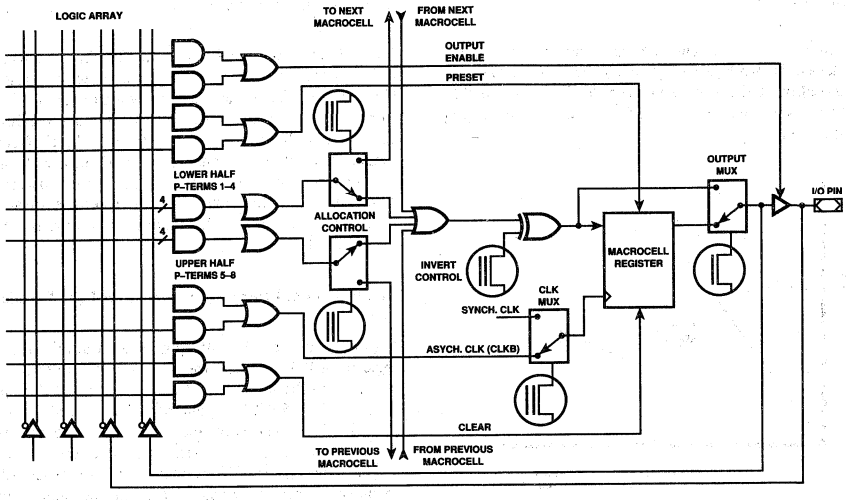


Figure 6. 5AC312 Macrocell Architecture

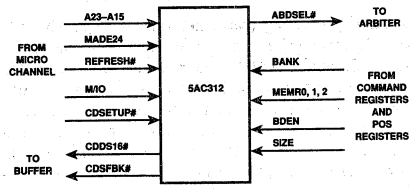


Figure 7. Pinout for Ethernet Adapter Decode Logic

Sequential Features

In addition to the allocatable OR architecture, the 5AC312 offers a host of other enhancements: separate register and pin feedback paths, register preset, and two product terms on clock, clear, preset and output enable control lines (Figure 6). These features, on top of the programmable AND/allocatable OR array, make the 5AC312 ideally suited for complex sequential designs.

Though not a state machine, the programmable baud rate generator circuitry for the Intel 8251 Universal Synchronous/Asynchronous Receiver/Transmitter does have a good mixture of gates and register functions, qualifying it for the state machine application region. The input baud rate (BAUDIN) is divided down to lower baud rates through a series of toggle flip-flops. Then, based on the select data stored in the D2, D1, D0 flip-flops, one of the divided-down baud rates is selected and sent out on the baud rate out pin (BAUDOUT).

Implementing the design in a standard 24-pin PLD (exemplified here by the Intel 5C060) is very costly. The data inputs must be latched inside a macrocell, using not only the macrocell but also the pin. The divide down toggle flip-flops cannot be buried, resulting in the loss of a pin for each flip-flop. The net utilization for the 5C060 implementation is 12 of 16 macrocells and 16 of 24 pins, virtually all of the device.

```
EQUATIONS:

AM1 =  A23 &   A22   & A21 &   A20 &    MEMR2 & !SIZE;   %  EXTENDED RANGE 32 K   %
AM2 =  A23 &   A22   & A21 &   A20 &    MEMR2 &  SIZE;   %  EXTENDED RANGE 64 K   %
AM3 =  !A23 &  !A22  & !A21 &  !A20 &   !MEMR2;          %  NOT EXTENDED RANGE    %

AM4 =  A19 &   A18   & !A17 &   !SIZE          &         %  EITHER, 32 K          %
      (!A16 #  !MEMR1 & !A15 &  !MEMR0          #        %  MATCH XC0000          %
       !A16 &  !MEMR1 & A15 &   MEMR0          #        %  MATCH XC8000          %
       A16 &   MEMR1 & !A15 &   !MEMR0         #        %  MATCH XD0000          %
       A16 &   MEMR1 & A15 &    MEMR0);        %  MATCH XD8000          %

AM5 =  A19 &
     ( (!A18 &   A17   &  !MEMR1   & !A16 & !MEMR0) #    %  EXTENDED, 64 K       %
       (!A18 &   A17   &  !MEMR1   &  A16 &  MEMR0) #    %  MATCH FA0000 , 64 K   %
       ( A18 &   !A17  &  !MEMR1   &  A16 &  MEMR0) #    %  MATCH FB0000 , 64 K   %
       ( A18 &   !A17  &  !MEMR1   & !A16 & !MEMR0) #    %  MATCH FC0000 , 64 K   %
       ( A18 &   !A17  &  !MEMR1   &  A16 &  MEMR0) );   %  MATCH FD0000 , 64 K   %

AM6 =  A19 &   A18   & !A17 &   SIZE &                   %  NOT EXTENDED, 64 K   %
     ( !MEMR0 &  !A16  #                                 %  MATCH 0C0000 , 64 K   %
       MEMR0 &   A16       );                            %  MATCH 0D0000 , 64 K   %

AM7 =  MADE24 & REFRESH_ ;                               %  ALL CYCLES            %

AM8 =  SETUP_  &  M_IO;                                  %  MEMORY CYCLES         %

AM9 =  !SETUP_ &  !M_IO;                                 %  SETUP  CYCLES         %

CDDS16 =  !BDEN &  !BANK &  AM7 & AM8  &     %  ENABLED, SRAM, NOT SETUP  AND  %
        ( AM1 &    AM4                 #     %  EXTENDED, 32             OR   %
          AM3 &    AM4                 #     %  NOT EXTENDED, 32         OR   %
          AM2 &    AM5                 #     %  EXTENDED, 64             OR   %
          AM3 &    AM5                 ) );  %  NOT EXTENDED, 64             %

CDSFBK =  !(BDEN &  AM7 &  AM8 &          %  ENABLED, NOT SETUP       AND  %
        ( AM1 &    AM4                 #     %  EXTENDED, 32             OR   %
          AM3 &    AM4                 #     %  NOT EXTENDED, 32         OR   %
          AM2 &    AM5                 #     %  EXTENDED, 64             OR   %
          AM3 &    AM5                 ) );  %  NOT EXTENDED, 64             %

ABDSEL =  !(AM1 &  AM4 &  AM7 & AM8  #     %  EXTENDED, 32             OR   %
          AM2 &    AM4 &  AM7 & AM8  #     %  NOT EXTENDED, 32         OR   %
          AM2 &    AM5 &  AM7 & AM8  #     %  EXTENDED, 64             OR   %
          AM3 &    AM6 &  AM7 & AM8  #     %  NOT EXTENDED, 64         OR   %
          AM7 &    AM9 );                  %  SETUP CYCLE                    %

END$
```

Figure 8. Micro Channel Decoder Nested Input Equations

Implementing the same design in the Intel 5AC312 uses a much smaller amount of space. By using the input latches available on the 5AC312, the select data inputs can be stored immediately at the input pin rather than inside a macrocell. This saves a macrocell, saves a pin, and decreases the delay time. Second, since the 5AC312 has separate register and pin feedbacks on each macrocell, the baud rate divider can be buried by using the register feedback paths while the input feedback paths remain available for use as standard inputs. Inside the 5AC312, the circuit consumes 8 of 12 macrocells, and 7 of 24 pins, a significant I/O pin savings.

In fact, the I/O pin savings is so significant that the accompanying address decode circuitry—which would be implemented typically in a 20L8 or second PLD—can be added to the 5AC312

```
EQUATIONS:
  ABDSEL =   (A23 * A22 * A21 * A20 * MEMR2 * SIZE * A19 + A18' * A17 * MADE
              REFRESH_ * SETUP_ * M_IO * A16' * MEMR1' * MEMR0'
           +  A23 * A22 * A21 * A20 * MEMR2 * SIZE * A19 * A18' * A17 * MADE24 *
              REFRESH_ * SETUP_ * M_IO * A16 * MEMR1' * MEMR0
           +  A23' * A22' * A21' * A20' * MEMR2' * A19 * A18 * A17' * MADE24 *
              REFRESH_ * SETUP_ * M_IO * A16' * MEMR1' * A15' * MEMR
           +  A23' * A22' * A21' * A20' * MEMR2' * SIZE' * A19 * A18 * A17' MADE24 *
              REFRESH_ * SETUP_ * M_IO * A16 * MEMR1 * A15' * MEMR0'
           +  A23' * A22' * A21' * A20' * MEMR2' * SIZE' * A19 * A18 * A17' MADE 24 *
              REFRESH_ * SETUP_ * M_IO * A16' * MEMR1' * A15 * MEMR0
           +  A23' * A22' * A21' * A20' * MEMR2' * A19 * A18 * A17' * MADE24 *
              REFRESH_ * SETUP_ * M_IO * A16 * MEMR1 * A15 * MEMR0
           +  A23 * A22 * A21 * A20 * MEMR2 * SIZE' * A19 * A18 * A17' * MADE24 *
              REFRESH_ * SETUP_ * M_IO * A16' * MEMR1' * A15' * MEMR
           +  A23 * A22 * A21 * A20 * MEMR2 * SIZE' * A19 * A18 * A17' * MADE24 *
              REFRESH_ * SETUP_ * M_IO * A16 * MEMR1 * A15' * MEMR0'
           +  A23 * A22 * A21 * A20 * MEMR2 * SIZE' * A19 * A18 * A17' * MADE24 *
              REFRESH_ * SETUP_ * M_IO * A16' * MEMR1' * A15 * MEMR0
           +  A23 * A22 * A21 * A20 * MEMR2 * A19 * A18 * A17' * MADE24 * REFRESH_
              * SETUP_ * M_IO * A16 * MEMR1 * A15 * MEMR0
           +  A23' * A22' * A21' * A20' * MEMR2' * SIZE * A19 * A18 * A17' * MADE24 *
              REFRESH_ * SETUP_ * M_IO * A16' * MEMR0'
           +  A23' * A22' * A21' * A20' * MEMR2' * SIZE * A19 * A18 * A17' * MADE24 *
              REFRESH_ * SETUP_ * M_IO * A16 * MEMR0
           +  A23 * A22 * A21 * A20 * MEMR2 * SIZE * A19 * A18 * A17' * MADE24 *
              REFRESH_ * SETUP_ * M_IO * A16' * MEMR1' * MEMR0'
           +  A23 * A22 * A21 * A20 * MEMR2 * SIZE * A19 * A18 * A17' * MADE24 *
              REFRESH_ * SETUP_ * M_IO * A16 * MEMR1 * MEMR0
           +  MADE24 * REFRESH_ * SETUP_' * M_IO');

  CDSFDBK =  (BDEN * MADE24 * REFRESH_ * SETUP_ * M_IO * A23 * A22 * A21 * A20 *
              MEMR2 * SIZE * A19 * A18' * A17 * A16' * MEMR1' * MEMR0'
           +  BDEN * MADE24 * REFRESH_ * SETUP_ * M_IO * A23 * A22 * A21 * A20 *
              MEMR2 * SIZE * A19 * A18' * A17 * A16 * MEMR1' MEMR0
           +  BDEN * MADE24 * REFRESH_ * SETUP_ * M_IO * A23' * A22' * A21' * A20'
              * MEMR2' * A19 * A18 * A17' * A16' * MEMR1' * A15' * MEMR
           +  BDEN * MADE24 * REFRESH_ * SETUP_ * M_IO * A23' * A22' * A21' * A20'
              * MEMR2' * SIZE' * A19 * A18 * A17' * A16 * MEMR1 * A15' * MEMR0
           +  BDEN * MADE24 * REFRESH_ * SETUP_ * M_IO * A23' * A22' * A21' * A20'
              * MEMR2' * SIZE' * A19 * A18 * A17' * A16' * MEMR1' * A15 * MEMR0
           +  BDEN * MADE24 * REFRESH_ * SETUP_ * M_IO * A23' * A22' * A21' * A20'
              * MEMR2' * A19 * A18 * A17' * A16 * MEMR1 * A15 * MEMR0
           +  BDEN * MADE24 * REFRESH_ * SETUP_ * M_IO * A23 * A22 * A21 * A20 *
              MEMR2 * SIZE' * A19 * A18 * A17' * A16 * MEMR1' * A15' * MEMR0'
           +  BDEN * MADE24 * REFRESH_ * SETUP_ * M_IO * A23 * A22 * A21 * A20 *
              MEMR2 * SIZE' * A19 * A18 * A17' * A16' * MEMR1' * A15 * MEMR0
           +  BDEN * MADE24 * REFRESH_ * SETUP_ * M_IO * A23 * A22 * A21 * A20 *
              MEMR2 * A19 * A18 * A17' * A16 * MEMR1' * A15 * MEMR0
           +  BDEN * MADE24 * REFRESH_ * SETUP_ * M_IO * A23' * A22' * A21' * A20'
              * MEMR2' * SIZE * A19 * A18' * A17' * A16' * MEMR0'
           +  BDEN * MADE24 * REFRESH_ * SETUP_ * M_IO * A23' * A22' * A21' * A20'
              * MEMR2' * SIZE * A19 * A18 * A17' * A16 * MEMR0
           +  BDEN * MADE24 * REFRESH_ * SETUP_ * M_IO * A23 * A22 * A21 * A20 *
              MEMR2 * SIZE * A19 * A18 * A17' * A16' * MEMR1' * MEMR0'
           +  BDEN * MADE24 * REFRESH_ * SETUP_ * M_IO * A23 * A22 * A21 * A20 *
              MEMR2 * SIZE * A19 * A18 * A17' * A16 * MEMR1 * MEMR0)

  CDDS16  =  (BDEN * BANK' * MADE24 * REFRESH_ * SETUP_ * M_IO * A23 * A22 * A21
              * A20 * MEMR2 * SIZE * A19 * A18' * A17 * A16' * MEMR1' * MEMR0'
           +  BDEN * BANK' * MADE24 * REFRESH_ * SETUP_ * M_IO * A23 * A22 * A21
              * A20 * MEMR2 * SIZE * A19 * A18' * A17 * A16 * MEMR1' * MEMR0
           +  BDEN * BANK' * MADE24 * REFRESH_ * SETUP_ * M_IO * A23' * A22' *
              A21' * A20' * MEMR2' * A19 * A18 * A17' * A16' * MEMR1' * A15' * MEMR0'
           +  BDEN * BANK' * MADE24 * REFRESH_ * SETUP_ * M_IO * A23' * A22' *
              A21' * A20' * MEMR2' * SIZE' * A19 * A18 * A17' * A16 * MEMR1 * A15' *
              MEMR0'

           +   BDEN * BANK' * MADE24 * REFRESH_ * SETUP_ * M_IO * A23' * A22' *
              A21' * A20' * MEMR2' * SIZE' * A19 * A18 * A17' * A16' * MEMR1' * A15 *
              MEMR0
           +  BDEN * BANK' * MADE24 * REFRESH_ * SETUP_ * M_IO * A23' * A22' *
              A21' * A20' * MEMR2' * A19 * A18 * A17' * A16 * MEMR1 * A15 * MEMR0
           +  BDEN * BANK' * MADE24 * REFRESH_ * SETUP_ * M_IO * A23 * A22 * A21
              * A20 * MEMR2 * SIZE' * A19 * A18 * A17' * A16' * MEMR1' * A15 * MEMR0'
           +  BDEN * BANK' * MADE24 * REFRESH_ * SETUP_ * M_IO * A23 * A22 * A21
              * A20 * MEMR2 * SIZE' * A19 * A18 * A17' * A16 * MEMR1' * A15 * MEMR0
           +  BDEN * BANK' * MADE24 * REFRESH_ * SETUP_ * M_IO * A23 * A22 * A21
              * A20 * MEMR2 * SIZE * A19 * A18 * A17' * A16' * MEMR1' * A15 * MEMR0
           +  BDEN * BANK' * MADE24 * REFRESH_ * SETUP_ * M_IO * A23 * A22 * A21
              * A20 * MEMR2 * A19 * A18 * A17' * A16 * MEMR1 * A15 * MEMR0
           +  BDEN * BANK' * MADE24 * REFRESH_ * SETUP_ * M_IO * A23' * A22' *
              A21' * A20' * MEMR2' * SIZE * A19 * A18 * A17' * A16' * MEMR0'
           +  BDEN * BANK' * MADE24 * REFRESH_ * SETUP_ * M_IO * A23' * A22' *
              A21' * A20' * MEMR2' * SIZE * A19 * A18 * A17' * A16 * MEMR0
           +  BDEN * BANK' * MADE24 * REFRESH_ * SETUP_ * M_IO * A23 * A22 * A21
              * A20 * MEMR2 * SIZE * A19 * A18 * A17' * A16' * MEMR1' * MEMR0'
           +  BDEN * BANK' * MADE24 * REFRESH_ * SETUP_ * M_IO * A23 * A22 * A21
              * A20 * MEMR2 * SIZE * A19 * A18 * A17' * A16 * MEMR1 * MEMR0);
```

Figure 9. Micro Channel Decoder Expanded and Reduced Equations

(Figure 11). The 14 address inputs (A13–A0), along with the memory or I/O status signal (M/IO) are fed into the PLD to generate the baud rate select data clock signal (BAUDSEL) and the 8251 Command/Data (C/D) and Chip Select (/CS) signals. The net result is a 10 of 12 macrocell, 24 of 24 pin, single-chip solution.

Based around a novel programmable AND/allocatable OR array structure, the Intel 5AC312 is uniquely suited to cover both highly combinational and complex sequential designs (Figure 12). The 5AC312 is made combinationally powerful through a 0 - 16 product term allocation arrangement and sequentially powerful through separate register and pin feedback and other features. The 5AC312's latched input capability is an asset in both combinational and sequential applications. The net result is a combinationally powerful, sequentially powerful 24-pin device.

ACKNOWLEDGEMENTS

Special thanks to David Poisner of the Intel Datacomm Focus Group in Folsom, CA and Prof. J. Michael Dunlap and Robert A. Miller of Willamette University in Salem, OR for the use of their designs in this paper.

REFERENCES

[1] An Engineering Approach to Digital Design, William I. Fletcher, Prentice-Hall Inc., 1980, pp. 276, 280, 281. (Provided as an example.)

[2] PAL is a registered trademark of Monolithic Memories Inc.

[3] Product Term Allocation is an Intel Patent Pending.

[4] Personal System/2 and Micro Channel are trademarks of International Business Machines Corp.

[5] Ethernet is a trademark of Xerox Corp.

[6] ESPRESSO is a copyright of the University of California at Berkeley.

[7] IBM PC/AT is a registered trademark of International Business Machines Corp.

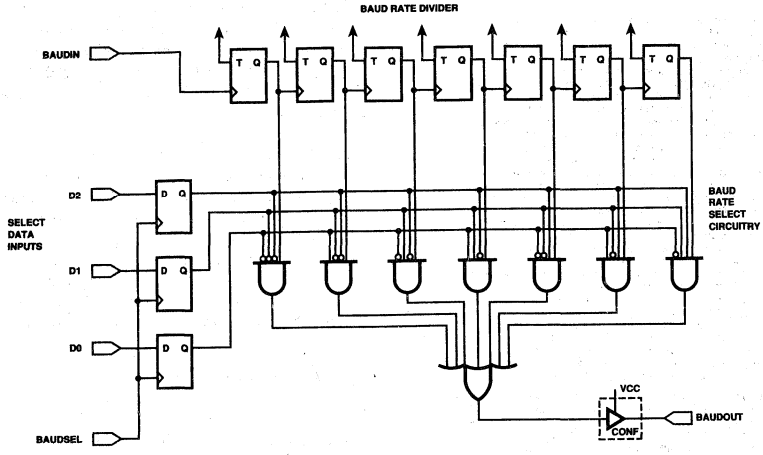


Figure 10. Programmable Baud Rate Generator Circuitry

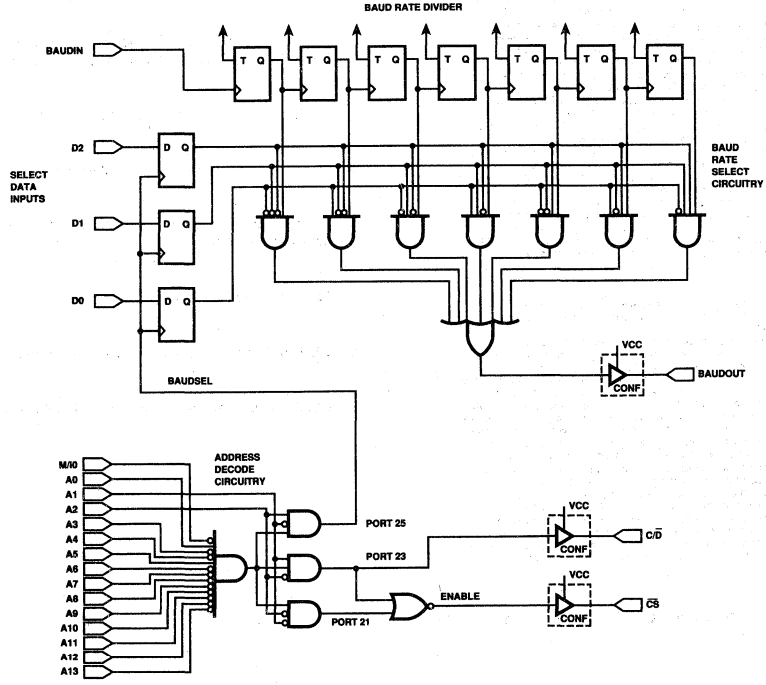


Figure 11. Programmable Baud Rate Generator

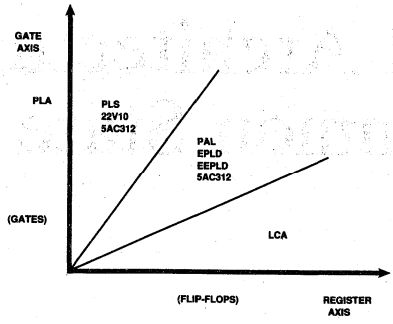


Figure 12. 5AC312 Application Areas

Advanced Architecture PLDs Solve Common State Machine Problems

by Liliyas S. Koumis
Technical Marketing Engineer

INTRODUCTION

The introduction of programmable logic devices (PLD) was a true revolution in the hardware design world. It enabled engineers to shrink circuits requiring several devices onto a single device thus simplifying their designs while saving space and power. Traditionally, PLDs have been used in combinational circuits such as address decoders as well as sequential circuits such as bus arbitration schemes. During the last few years, advances and improvements in PLD architectures enabled the devices to grow more complex while addressing the never-ending quest for higher density and faster speeds. Despite these improvements, engineers still face certain problems and limitations when implementing state machine designs with PLDs. The Intel 5AC312 and 5AC324, multipurpose generic erasable PLDs, offer a solution to these problems that gives engineers a better device to implement their designs.

A typical programmable logic device is composed of a user-programmable AND array, a fixed OR gate, followed by an output register which includes a feedback path from the output to the programmable AND array. Combination of these elements is commonly referred to as a 'macrocell.' The existence of a feedback path from the output registers to the AND array makes PLDs ideal candidates for state machine implementations.

TYPES OF STATE MACHINES POSSIBLE IN PLDs

The three basic categories of state machines are Class A, Class B and Class C, better known as MEALY, MOORE TYPE A and MOORE TYPE B respectively. It is possible to implement any of the classes of state machines in a PLD, however the efficiencies vary with state machine class. The main characteristic of the Class A machine is that its outputs to the external world are a function of both the input and the present state of the machine as shown in Figure 1. Mathematically, this can be expressed as:

$z^{n+1} = f(x^n, y^n)$

Class A Machine
(Mealy Machine)

where 'z' is the decoded output, 'x' is the input, 'y' is output of the next state decoder and 'n' is the present state.

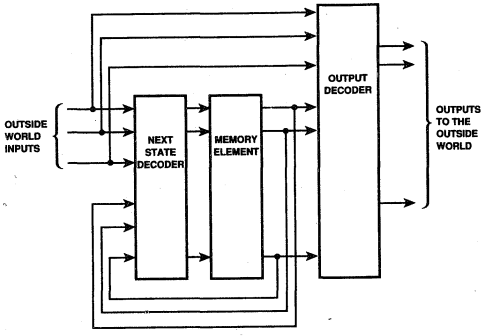


Figure 1. Class A State Machine

The Class B machine differs from Class A in that its output is only dependent on the present state of the machine through output decoding, or better expressed as:

$z^{n+1} = f(y^n)$

Class B Machine
(Moore Type A Machine)

Finally, the Class C macine is essentially the same as the Class B but requires no output decoding: the outputs *are* the next state of the machine:

$z^{n+1} = y^{n+1}$

Class C Machine
(Moore type B Machine)

If a Class A or a Class B machine is implemented in a standard PLD, two macrocells would be required per state; one macrocell

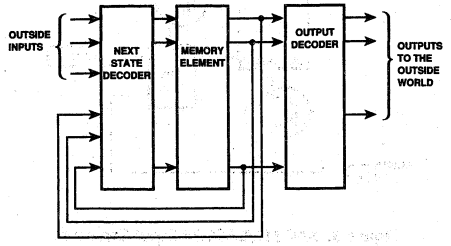


Figure 2. Class B State Machine

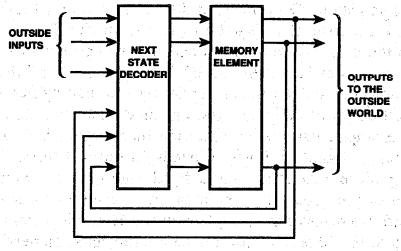


Figure 3. Class C Machine

has to be used for the input decoding and the other for the output decoding. The use of an extra macrocell for output decoding is not an efficient use of the device resources. For example, the largest state machine that can be implemented in an eight-register PLD is one with only four states variables. On the other hand, since a Class C machine does not perform output decoding, only one macrocell per state is required. As a result, that a machine with eight state variables could easily fit in the same eight-register PLD.

PROBLEMS WITH STATE MACHINE DESIGNS IN PLDs

Despite the architectural advantages of implementing a Class C machine, traditional PLDs still inherit serious timing problems and structural shortcomings.

The first problem is violation of setup and hold times of the output registers. This is encountered commonly in environments where the inputs are asynchronously changing with the device clock. Sources for these problems may be different data paths for each input or origination of these input signals from circuits that use a different clock than that used for the output registers. This kind of problem causes the output to glitch and may cause the machine to enter an invalid or incorrect state. The problem traditionally has been solved by adding external metastable-hardened registers to synchronize the inputs with the outputs. This solution, however, has obvious drawbacks such as an increase in chip count, additional time delays, and an increase in power consumption. In some cases, an eight-register IC is added even when only a few inputs must be synchronized.

The second most common problem is missed input pulses of short duration. In modern and complex circuits, short pulses may arrive at the inputs and disappear at a faster rate than the PLD clock. This causes these inputs to be missed by the PLD, which in turn causes erroneous operation of the state machine. Traditional PLDs offer no solution to this problem. Engineers have had to resort to adding circuitry to ensure that the inputs are present long enough to be seen by the PLD clock. This additional circuitry further complicates designs, adds delays, increases power consumption and adds an unnecessary burden to the engineers.

Finally, the most frustrating problem for engineers is that the number of product terms available per macrocell is fixed. Engineers usually assign the states to the output pins randomly, write the equations and allow the software to determine whether the equations fit their chosen device. If the number of product terms required to implement the given equations is greater than the fixed number of available product terms, the designer devotes additional time and resorts to more 'innovative' approaches, such as breaking down the state machine into smaller parts to fit the device. This not only introduces additional time delays and reduces the effective use of the device, but it also increases development time and board cost.

SOLUTIONS TO THESE COMMON PROBLEMS

The above discussion illustrates the need for a more sophisticated generation of Programmable Logic Devices that address these problems. Two new Intel PLDs, the 5AC312 and 5AC324, are specifically targeted at fulfilling the requirements of better set-up and hold timing, faster input clock rate and flexible product terms. The AC in the part name stands for "Advanced CMOS," the 3 stands for third generation and 12/24 is the number of macrocells in each device. This family of Intel Erasable Programmable Logic Devices uses advanced CHMOS* EPROM cells instead of polysilicon fuses as a logic control element. This process enhances the testability and reliability of the devices while significantly reducing power consumption. For the remaining portions of this paper, the 5AC312 will be referred for ease of reference, but for all practical purposes, the 5AC324 is functionally identical to the 5AC312 but contains twice the number of macrocells and ten register/latched inputs instead of eight.

As it can be seen in Figure 4, the 5AC312 has the architectural features of a Class C machine but also offers additional features address the issues discussed earlier. The input structure of the 5AC312 offers several programmable options, each addressing a particular need or problem.

To address the first problem-violation of setup and hold times of the output registers—the 5AC312 offers an additional register/latch input with a programmable clock. The clock can be the same as the output register clock shifted by 180°, a separate high frequency clock, or be generated by a product term from the logic array.

By using the first clock option, synchronization is achieved, and thus the risk of output glitches is minimized. By cascading the input and output registers and shifting the input clock 180° from the output register clock, an additional advantage is gained by

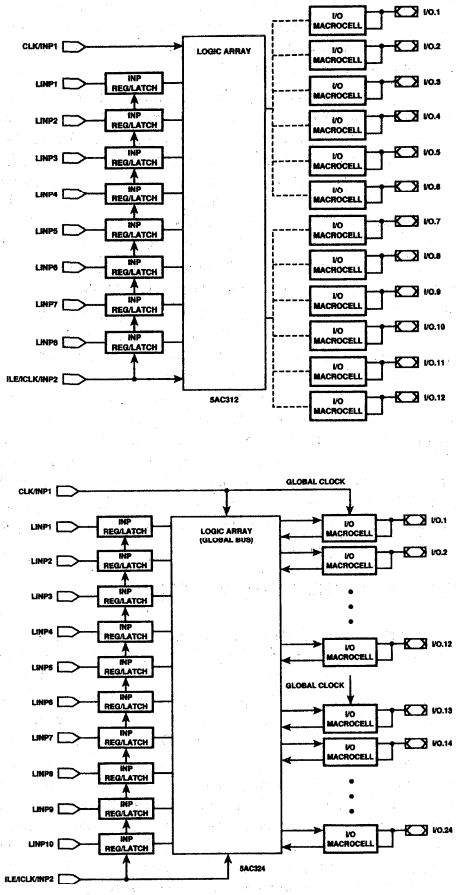


Figure 4. Architecturally Advanced 5AC312 and 5AC324 Global Block Diagrams

allowing enough time to satisfy the setup and hold time requirements of the output registers. Metastability characteristics of the device is of particular concern and are discussed later in this paper.

The second option, a separate high frequency clock, enables the device to sample inputs of very short time duration. This clock operates up to 50 MHz, with an input register setup of only 5 ns. If the latch feature is selected, the setup time is reduced to 0 ns. Of course, a mode can be selected where the input data flows through, bypassing the register/latch combination. The third clock option, clocking the input registers with the output of a product term from the logic array, is ideal for applications where registers are to be clocked only when a certain input condition is met.

To address the fixed product term problem, the 5AC312 implements an innovative solution called 'product term allocation.' In

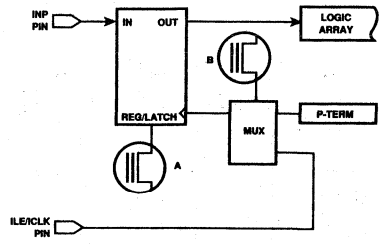


Figure 5. 5AC312/5AC324 Input Structure

each individual macrocell, the eight product terms are sub-divided into two groups of four. The product term allocation is achieved by allowing each of these four product term groups to be borrowed from or lent to adjacent macrocells. By 'allocating' product terms between adjacent macrocells, any register can be driven by as many as 16 product terms by borrowing unused product terms from its neighbors. Conversely, as little as zero product terms can be used if a macrocell lends to both of its neighbors. The 12 macrocells in the device have been divided into two groups of six each (or two groups of 12 for the 5AC324) called the "rings" that help define adjacent macrocells for borrowing and lending purposes.

The efficiency of this configuration can be demonstrated with a simple example. Assume an excitation function needs four product terms and another function needs 10 product terms. Implementing these functions with a fixed eight-product-term PLD requires one macrocell to implement the four-product-term function, and two macrocells to implement the 10 product terms function. To fit the 10 product term function, the equation needs to be broken into two parts, thereby increasing the delay. Therefore, out of 24 available product terms (three groups of eight), 14 are used (14/24 = 58% efficiency). Using the 5AC312 to implement the same functions yields the following: the four-product-term function is implemented with half of a macrocell, allowing the other half to be allocated to the adjacent macrocell for implementing the 10-product-term function. No design-splitting is required. Therefore, out of 16 product terms, 14 are used. This translates to 88% product term utilization. The product term allocation is completely transparent to the user since it is achieved through software. When the compiler determines that an additional number of product terms is required, it automatically allocates resources to fit the required excitation function.

METASTABILITY CHARACTERISTICS

Although metastability is a relatively rare event, ignoring it can cause serious timing problems. The input registers found in the 5AC312 offer excellent recovery time where metastability is of concern. Metastability can be simply described as the inability of a register to decide the state of its output within a fixed amount of time. This event usually occurs when synchronizing an external event with a periodic clock. If a flip flop is clocked nearly at the same time as changing data, there is a small window of time where the output of the register is unknown. This window of time is the recovery time (t_{re}) of the flip-flop and is typically in the order of nano-seconds. Designers at Intel have performed tests to obtain the recovery time for the 5AC3xx family of devices and have concluded that the Intel devices have better recovery times

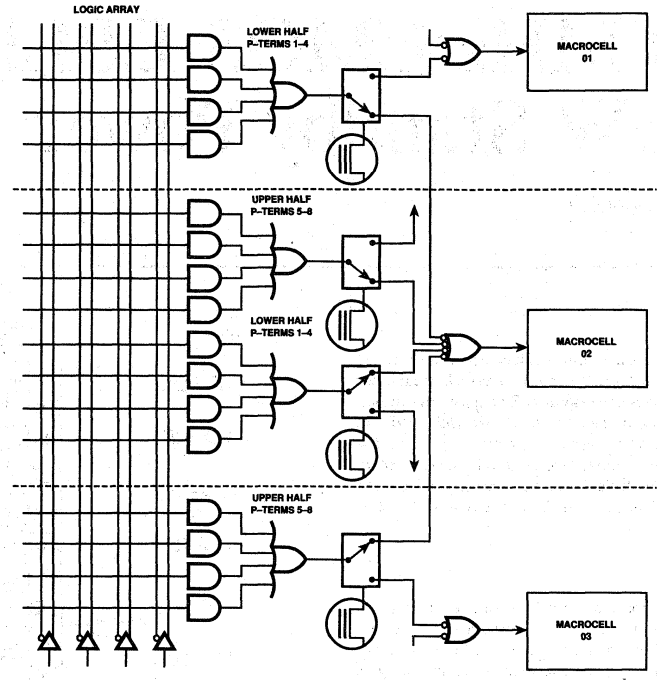


Figure 6. Product Term Allocation

than familiar TTL devices such as 74F74. Table 1 shows sample data taken at a clock frequency of 2 MHz and data frequency of 1 MHz.

Additional information on the procedure used to obtain this data and its use can be found in references one and three.

CONCLUSION

Because of their internal architectural characteristics Programmable Logic Devices have become the ideal method to implement state machine designs. This is evident by the wide variety of applicatons where programmable logic devices are found today. The latest generation of PLDs, with the advantages of programmable I/O pins and expanded number affixed product terms, are certain to replace traditional off-the-shelf logic as designers discover their usefulness in modern applications. However, to overcome the problems associated with setup and hold time violations, missed inputs and fixed number of product terms, a new generation of PLDs was needed. The Intel 5AC312 and 5AC324 overcome these problems by providing selectable input register/latch option with excellent metastability characteristics and allocatable product terms.

Table 1

Device	Recovery Time (ns)
7474	1.6
74LS74	1.5
74S374	0.91
74F373	0.70
74F74	0.40
5AC312/5AC324	0.35

*CHMOS is a patented process of Intel Corporation.

REFERENCES

[1] Chaney, Thomas J., "Measured Flip-Flop Responses to Marginal Triggering," IEEE Transaction on Computers, Vol. C-32 No. 12, December 1983.

[2] Fletcher, William I., "An Engineering Approach to Digital Design," Prentice-Hall Inc., 1980.

[3] Stoll, Peter A., "How to Avoid Synchronization Problems," VLSI Design, November/December 1982.

[4] Weigl, Karl H., "Eliminating Common Problems in State Machine Designs Using Innovative PLD Architectures," SOUTHCON, December 1987.

A 150 MHz CMOS EPLD with μW Standby Power

by Terry L. Baucom and Michael J. Allen

ABSTRACT

This paper reports on the design and performance of a universal 300-gate equivalent CMOS EPLD. Innovative circuit techniques were utilized to achieve high performance. The maximum propagation delay in combinatorial mode is 6 nanoseconds, while in register mode, the device can operate at frequencies up to 150 megahertz. The standby and active ICC is 10 microamps and 70 milliamps respectively. The device was fabricated on a 1.0-micron double-metal CMOS EPROM process. The die size is 1.95 × 2.05 millimeters.

INTRODUCTION

In recent years, CMOS programmable logic devices have made great strides in narrowing the historical performance gap that has separated them from their bipolar counterparts. This paper describes a new CMOS EPLD that succeeds in closing the gap, while preserving the CMOS benefits of low power dissipation and testability. Innovative circuits and prudent layout techniques were used to achieve register mode operation up to 150 megahertz. The active ICC is 70 milliamps and the standby ICC is 10 microamps. The device was fabricated on a 1.0-micron double-metal CMOS EPROM process. The size is 2 millimeters square.

APPLICATION

This device is a general purpose, 20-pin programmable logic device and is capable of replacing most standard 20-pin programmable devices. [1] This device was designed for applications requiring low complexity (~300 gates) and high performance. For example, this device will satisfy the performance requirements of the bus state tracker for a 33 MHz 386™ microprocessor system, which must run at twice the processor clock frequency. Due to the fast setup and clock to output performance of this device the performance goals of this application were satisfied with more margin than available with other solutions. Another key attribute of register mode operation is the superior metastability characteristics of the registers. This device has also been used for combinatorial applications to replace high performance bipolar logic devices on an EISA system board. The result was a board temperature reduction of 27 degrees Celsius with no performance penalty. In this application, the primary benefit provided by this device was high performance with lower power consumption and less heat dissipation. The performance was equal to that of the bipolar devices, while the power dissipation was lower by a factor of three.

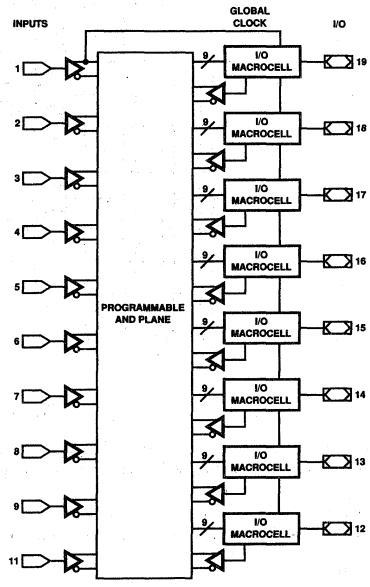


Figure 1.

CHIP ARCHITECTURE

The device architecture is based on a universal 20 pin format, with 10 dedicated input pins and 8 I/O pins. All input and I/O signals are global inputs to the programmable AND plane. Figure 1 shows a block diagram of the device. The product term output from the AND plane provides input signals to eight identical macrocells. The array provides 9 inputs to each macrocell: 1 for OE control and 8 to produce a sum of products signal. Each macrocell can individually be configured to operate in combinatorial or register mode. The source of the I/O feedback signal from each macrocell is established by the macrocell configuration, with pin feedback for combinatorial mode and register feedback for register mode. Figure 2 is the macrocell diagram. Each macrocell

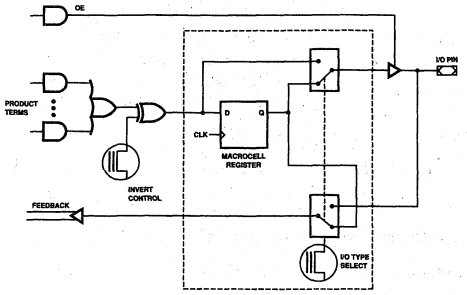


Figure 2.

contains a programmable invert option for the sum of products signal. A synchronous clock signal globally drives the registers in all eight macrocells. A reset signal is generated at power up to initialize the registers. A programmable global power down option is also available. When the power down option is used, the device will enter standby mode after 50 nanoseconds of inactivity on all pins. When the device is in stand by mode ICC drops to 10 microamps.

CIRCUITS AND PERFORMANCE

Several innovative circuits were developed to achieve the high performance of this device. The inputs feature a dual translator circuit for the true and complement wordlines. This provides individually optimized wordline switching speeds. The row drivers, shown in Figure 3, have been removed from the array pitch size limitations and are equipped with high voltage power switches for isolation during programming. The sense amplifier, Figure 4, makes use of a variable feedback reference and current-limiting scheme to limit the bitline voltage swing to 150 millivolts and to reduce pattern sensitivities. The delay between row driver input and sense amplifier output is only 2 nanoseconds.

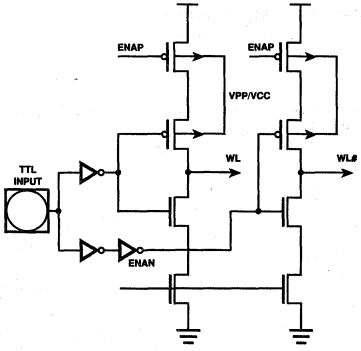


Figure 3.

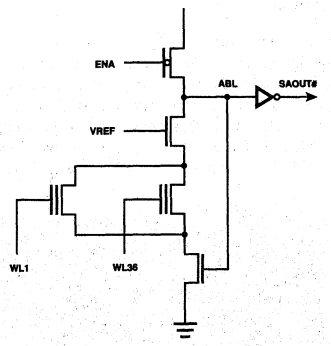


Figure 4.

A new circuit, Figure 5, was developed to OR the product terms, implement an invert option, and provide a latch for standby mode all in one delay stage. The output section of the macrocells contains a high speed multiplexing circuit. All of these circuits taken together reduce to three the number of logic stages from the sense amplifier to the output buffer. The output buffer was also designed for optimum performance as well as low noise. The output buffer uses an N-channel pullup device with a weak P-channel device in parallel to provide full CMOS output levels. A register with distributed TTL translator circuits and direct drive output provides very fast clock to output speeds. This design reduces the total number of stages from clock input to output buffer to two, and yields a 2.5 nanosecond clock to output delay. Total propagation delay in combinatorial mode is 6 nanoseconds. Register mode applications can be supported at frequencies up to 150 megahertz with active ICC of 70 milliamps. Figures 6 and 7 are scope photos of combinatorial and registered performance, respectively. A built-in power down mode allows the device to enter standby after 50 nanoseconds of inactivity, which reduces ICC to 10 μA. Table 2 is a summary of device characteristics.

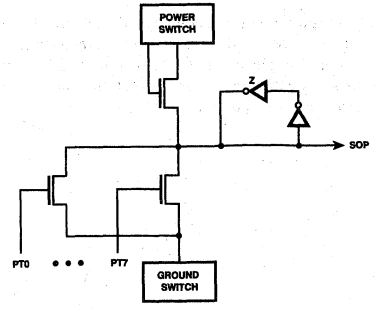


Figure 5.

PROCESS PARAMETERS

The device has been fabricated on a 1.0-micron N-well CMOS EPROM technology, that provides two polysilicon, and two metal interconnect layers. Tungsten silicide is used to reduce the polysilicon interconnect resistance. A summary of process parameters is provided in Table 1. Programming circuitry is implemented with high-voltage P-channel devices. The cell is a conventional double poly FLOTOX with a 9.0 × 4.7-micron cell size.

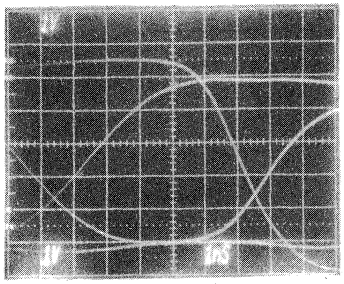


Figure 6.

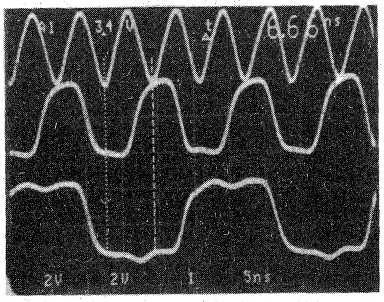


Figure 7.

CONCLUSION

The general purpose programmable logic device described here is an ideal choice for very high performance state-machine designs and combinatorial applications, such as todays 32-bit microprocessor designs. Its excellent power dissipation characteristics are an added advantage for todays tight board layouts.

Table 1. Process Parameters

Technology	1.0μ CMOS EPROM Double-level poly with silicide Double-level metal
N-channel Leff	0.7 micron
P-channel Leff	0.7 micron
Tox	250Å
Polycide resistivity	4 ohms/sq
Polysilicon pitch	2.5 microns
Metal-1 pitch	3.0 microns
Contact cut	1.2 × 1.2 microns
Metal-2 pitch	3.6μ
Via cut	1.2 × 1.2 microns

Table 2. Device Characteristics

Die Size	1.95 mm × 2.05 mm
Cell Size	9.0 × 4.7 microns
Fcnt	150 megahertz
Input setup time	3 nanoseconds
Clock to Output	2.5 nanoseconds
Active Icc (150 MHz)	70 milliamps
Standby Icc	10 microamps
Package	20-pin 300-mil dip 20-pin PLCC

ACKNOWLEDGEMENTS

We would like to thank Bob Casper, Reo Gargovich and Jim Negrey for their excellent layout support, Dan Smith, Kristin Bailey and Chris Wawro for their constructive review, and Intel Fab personnel for producing the silicon.

REFERENCES

[1] Monolithic Memories, Inc. "Programmable Array Logic Handbook," Chapter 1, 1984

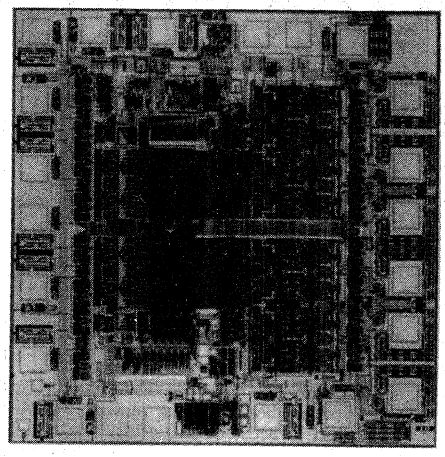


Figure 8.

A 6 ns CMOS EPLD with μW Standby Power

by Michael J. Allen

ABSTRACT

This paper describes a 6 nanosecond (ns) CMOS EPLD optimized for memory-address decoding applications, with a 1.0-μA standby ICC using a 1.0-micron N-well double layer metal EPROM process.

INTRODUCTION

Current generations of programmable logic devices fall into two general classes; either high speed, high power bipolar or slower, lower power CMOS devices [1,2]. By optimizing the architecture to fit the application area, it is possible to close the performance gap while maintaining CMOS power levels. This paper will describe a 28-pin CMOS EPROM-based programmable logic device optimized for memory address decoding applications, and using a novel architecture to provide high speed operation at CMOS power levels. Reprogrammability and 100% testability of EPROM technology are added benefits. Active power is less than 25% of slower bipolar solutions, and die area is 74 mils square.

APPLICATION

Overall performance of microprocessor systems starts with the speed of the system's microprocessor, but does not end there. System performance also depends on the ability of the memory and I/O subsystems to keep the processor running at full speed. A critical link in the speed path between the processor and memory is memory address decoding. If this function can be sped up, slower and more inexpensive memory devices can be used without sacrificing system throughput. For many pipelined applications, this data must also be latched. Integrating both decode and latch functions in one device reduces unnecessary buffering delays and allows greater performance.

CHIP ARCHITECTURE

The device is organized as 16 dedicated inputs feeding the programmable decode array, with 8 latched outputs controlled by a global Clock pin, as shown in Figure 1. Integrating the latch function onboard allows the device to be used in high performance applications with pipelined or multiplexed address/data buses. The device has a relatively small number of programmable elements, so device performance could be optimized at the expense of cell area. To maximize performance, programmed data is loaded into shift registers at power up and used to steer the inputs through the array. In this way the programmable element is not in the speed path, and full CMOS circuitry can be used.

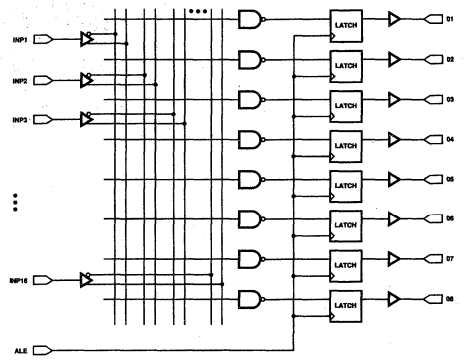


Figure 1.

CIRCUITS AND PERFORMANCE

Several innovative circuits were developed to allow high speed operation. To work in a TTL environment, the device must accept TTL compatible inputs, and supply CMOS levels for internal operation. It must also allow either true or complement input data to be connected to the decode function. In most present designs, this is accomplished by first running the input through a TTL/CMOS level translator, then inverters and buffers to drive both true and complement data through an array of programmable elments which control the connections to the decode function [3]. Multiple logic stages and interconnection delays limit the speed at which this circuitry can run, and the programmable element is in the speed path. Figure 2 shows a distributed TTL/CMOS buffer and integrated mux that removes much of this delay. The steering bits stored in the shift registers connect either the true or complement input data to the decode, through their own distributed TTL/CMOS translator. Disconnected paths are power switched to eliminate unnecessary Icc. The decode is implemented by a full CMOS tree decode. This circuit takes advantage of the fact that only the true or complement input is required at each individual decode, and requires only 16 inputs rather than the full 32.

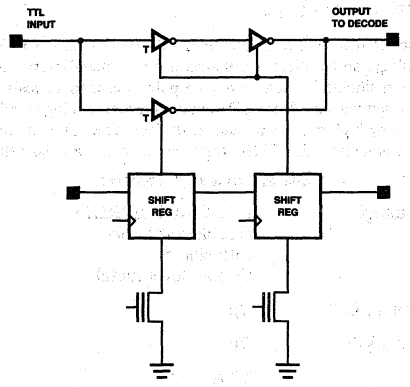


Figure 2.

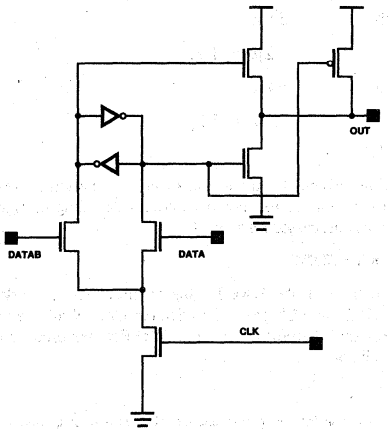


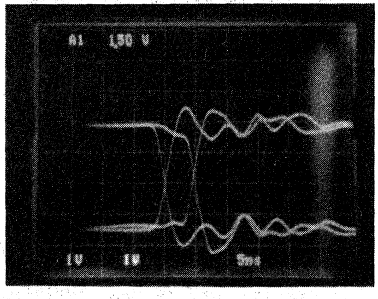
Figure 3.

The outputs feature a high speed direct drive latch with distributed TTL/CMOS translator, and symmetrical n-channel output, shown in Figure 3. This circuit has several advantages over traditional implementations. The distributed TTL translator integrated into the symmetrical latch allows a very fast turn off time for the output driver, and eliminates most crossover current by preventing the opposing device turn on until turn off is almost complete. The n-channel pullup contributes to reduced power supply noise by self limiting the pullup on low to high transitions, and allowing a reduced output voltage swing for TTL compatible systems by the use of an external load resistor. CMOS levels can be attained by removing the load resistor and allowing the weak P-channel to complete the pullup to full Vcc. A series resistor minimizes latchup trigger currents back into the output. The circuit reduces the number of logic stages required from clock input to output from 7 or more in conventional designs to just 3, and yields a Tco of 3.5 ns. The flow-through logic path is reduced to 7 inversions with a Tpd of 6 ns at 80°C and Vcc of 4.7 volts. Figure 4 shows an oscillograph of typical device performance. Standby Icc is less than 1 μA, and active current is less than 15 mA at 50 MHz.

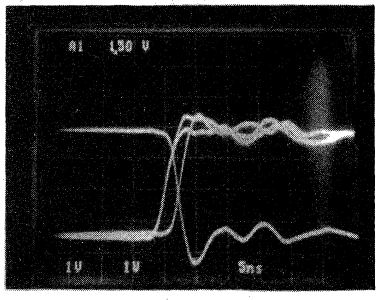
Device characteristics are presented in Table 1.

Table 1. Device Characteristics Vcc=4.7V Ta=80°C CL=30 pF

Die Size	1.5 mm × 2.4 mm
Cell Size	3.6μ × 3.8μ
Input to Output	6 ns
Clock to Output	3.5 ns
Active Icc (50 MHz)	15 mA
Standby Icc	1 μA
Package	28-pin .300 dip 28-pin PLCC
Technology	N-well 1.0μ CMOS EPROM wtih double layer metal



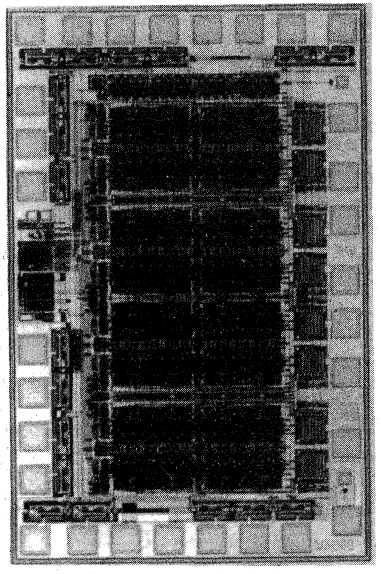
Tpd



Tco

TESTABILITY

Special circuitry allows for the device to be dynamically reconfigured by use of a serial shift mode. This allows for greatly simplified testing after assembly because the full speed path of the device can be exercised quickly without repeatedly programming and erasing the memory array. It also allows full speed testing of plastic OTP devices which cannot be erased after assembly. No high voltage signals are required to use this mode, which simplifies the test hardware. Test modes are built in to guarantee high programming margins and device reliability, and allow access to every cell location. Cell margins can be checked before and after programming, as well as after any accelerated temperature stresses, to identify any defective cells. Typical programming time is less than 100 μs, and cell margins are greater than 7 volts.



Die Photo

PROCESS PARAMETERS

The device has been fabricated in a 1.0μ N-well CMOS EPROM technology, and utilizes 2 polysilicon and 2 metalization layers. Tungsten silicide is used to reduce poly interconnect resistance. Table 2 summarizes Process Parameters. Programming circuitry is handled by high voltage p-channel devices. The cell is a conventional double poly FLOTOX structure with a 3.6 × 3.8μ cell size.

Table 2. Process Parameters

Technology	1.0μ CMOS EPROM Double-level poly with silicide Double-level metal
N-channel Leff	.7μ
P-channel Leff	.7μ
Tox	250Å
Polycide resistivity	4 ohms/sq
Polysilicon pitch	2.5μ
Metal-1 pitch	3.0μ
Contact cut	1.2μ × 1.2μ
Metal-2 pitch	3.6μ
Via cut	1.2μ × 1.2μ

CONCLUSION

The combination of high speed and low power dissipation makes this device an ideal choice for memory address decoding in high performance microprocessor systems [4].

ACKNOWLEDGEMENTS

I would like to thank Mike Love for his excellent layout work, Greg Ledenbach, Duane Chinnow, Ron Swartz, Chris Wawro and Abid Asghar for their helpful review, and Intel Fab personnel for producing the silicon.

REFERENCES

[1] Monolithic Memories, Inc. "Programmable Array Logic Handbook," Chapter 1, 1984.

[2] Altera Corporation, "EPLD Handbook," 1985.

[3] S.C. Wong, et al, "CMOS Erasable Programmable Logic Device with Zero Standby Power," ISSCC Digest of Technical Papers, Feb. 1986 pp 242-243.

[4] Intel Corporation, "Microcomputer Programmable Logic Handbook," 1989.

Optimized PLD Architectures for High Speed System Design

by Liliyas Sahba Koumis
Applications Engineer

INTRODUCTION

In recent years, many advances have been made in microprocessor capabilities and technology. The transition has been made from 8- and 16- to 32-bit wide architectures. An associated transition has been made in performance, with clock frequency rates now at 33 MHz or higher.

To maximize performance of systems that use these processors, new and faster memory designs are being employed. These include pipelining, interleaving and bursting memory access. All of these modes demand high speed memory for zero-wait state performance. They also require high speed microprocessor to memory logic.

This paper describes three new high speed CMOS programmable logic devices designed for implementing microprocessor to memory interfaces. The logic is optimized for performance in addition to offering low power consumption and high quality of CMOS-EPROM based technology.

MICROPROCESSOR TO MEMORY LOGIC

Conceptually, memory interface logic can be split into three categories (Figure 1) as follows:

1. Bus state tracking logic which follows the microprocessor's control signals to determine when and how the memory will be accessed.

2. Memory control logic which derives write, read and timing control signals for the memory (and signals to the microprocessor).

3. Address decoding to physically select an area of memory.

Each of these areas is now examined.

BUS STATE TRACKING

State tracking logic decodes microprocessor control signals to produce state information for memory control (Figure 2). The states inform the memory controller *IF* memory is being accessed (decodes memory/IO select signal from processor and memory select signal from decoder); *HOW* memory is being accessed (write/rd from processor), and *WHEN* it is being accessed (address strobes, bursting signals, and ready signals). Memory control logic uses this information to create memory access and data buffer control signals at the appropriate times.

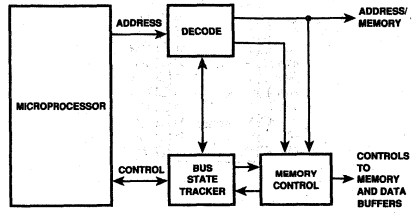


Figure 1. Memory Interface Logic

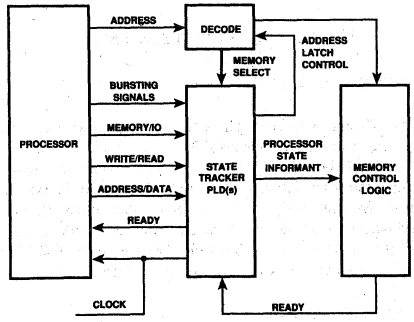
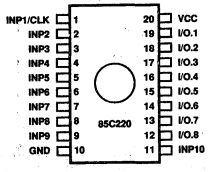


Figure 2. Typical Bus State Tracking Function

The required speed for state tracking logic is based on the operational speed of the processor's bus. Typically, this is a rate equal to or higher than the rated frequency of the microprocessor. For the Intel 386™ 33 MHz, the processor's bus operates at 66 MHz. Therefore the associated logic must also run at this speed. Since bus cycles are very short at high frequency, ideally the decoded bus states are passed on to memory control logic as quickly as possible.

The 85C220 CMOS PLD is designed to operate at high frequencies and produce very fast transfer of state information (Figure 3). Optimized for clocked operation, the 85C220 is rated at 80 MHz for externally connected signals and at 100 MHz for internal state machines. It achieves 80 MHz with a clock to output (t_{CO}) delay of under 5.5 ns and a data to clock set-up time of under 7 ns.

The 20-pin architecture allows for up to 8 outputs with each macrocell configured in either a clocked or combinatorial logic mode. Each logic output can be made of up to eight "sum-of-products" terms with a 9th term available for 3-state output control. The device offers a higher speed, lower power, more flexible alternative to commonly used D & E series PALs and GALs.



- SET-UP TO CLOCK — t_{SU} = 7ns
- CLOCK TO OUTPUT — t_{CO} = 5.5ns
- STATE MACHINE FREQUENCY — $\frac{1}{(tSU + tCO)}$ = 80MHz
- POWER SUPPLY CURRENT (80MHz) — ICC = 55mA

Figure 3. 85C220 Summary

MEMORY ACCESS LOGIC

The 85C220 can also be applied to memory control circuits. An example is the non-interleaved DRAM controller as used on the i860™ EV-AT board (Figure 4). A combination of high-speed state machine logic for RAS & CAS, refresh and decode logic is required. In this design, sub-10 ns combinatorial speeds and clocked operational speeds at greater than 58 MHz are required to produce the pipelined cycles and make use of the DRAM static column access for zero-wait state read operations.

ADDRESS DECODING

One of the functions directly in the speed path for memory access is the decoding of processor addressing. In a pipelined system, the decoded addresses will also need to be latched. Ideally, the decode and latch time will be minimized to allow slower, less expensive memory devices to be used.

The Intel 85C508 was designed with these requirements in mind. It employs a very fast, simple decoding scheme (Figure 5) followed by a high speed latch. Sixteen inputs can be decoded into eight memory enables and latched. The high speed of the latch to output (4.5 ns) and total propagation delay through the device (7.5 ns) performs the decode/latch tasks optimally.

INTEGRATED 80960 PLD: THE 85C960

In embedded systems, minimal chip count, system size and power consumption requirements will be goals along with high performance. By integrating the state tracking, memory control and address decoding logic, these goals can be achieved.

The 85C960 EPLD was designed as an optimal companion chip to track the 80960K series burst-bus and provide high speed control and decoded address signals to memory. A combination of fixed and programmable logic was used to collapse the equivalent of five high speed PALs and one PROM into a single chip (Figure 6). All timing parameters are set to 25 MHz with power reduced to ¼ watt.

IMPORTANCE OF CMOS EPROM TECHNOLOGY

Each of the devices described in this paper uses CMOS EPROM technology. While optimized for system speed, the CMOS nature of these parts also serves an important role in reducing the power consumption and heat dissipation versus the alternatives of bipolar PALs or competing CMOS devices. Each device is rated at a maximum of under 55 mA at full rated operational frequencies. Under typical conditions, the power consumption will be much lower. At these power levels the 85C508, 85C960 and 85C220 consume approximately one-fourth the power of PALs and half the power of competing CMOS. The lower power translates into lower system heat, thereby increasing reliability.

By employing EPROM technology high quality is attained. Each part is completely programmed, tested and erased before packaging to ensure complete functionality. The resulting quality levels, even in one-time-programmable plastic packages is typically under electrical 500 DPM and achieves programming yields greater than 99.9%.

CONCLUSIONS

Traditionally, high speed microprocessor to memory interfaces required the use of bipolar PALs to meet performance needs. New high speed CMOS devices are now available that are optimized for meeting important system performance criteria. These CMOS devices provide very high speed state machine and decode capability, while reducing system power, reducing system heat, and providing increased quality and reliability.

386 and i860 are trademarks of the Intel Corporation.
PAL is a trademark of AMD/MMI Corporation.
GAL is a trademark of Lattice Semiconductor Corporation.

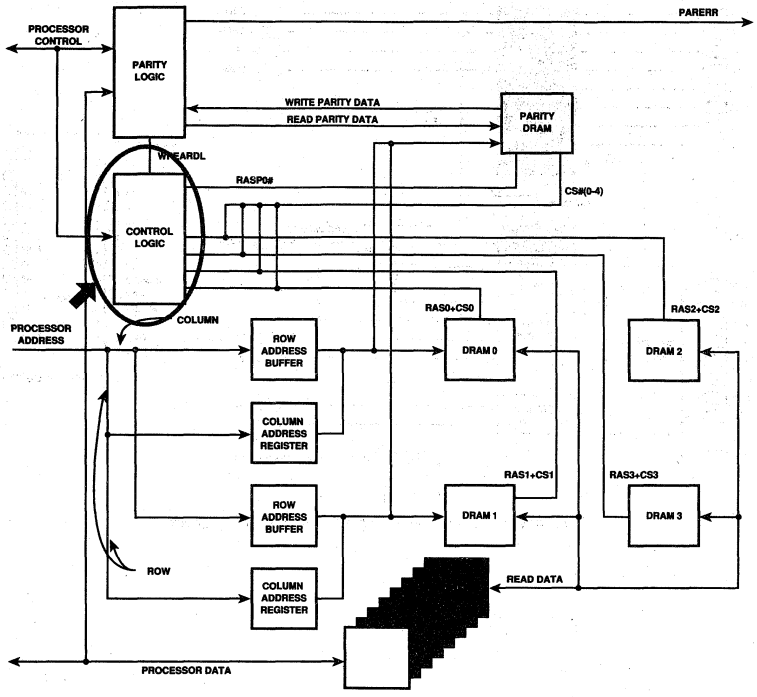
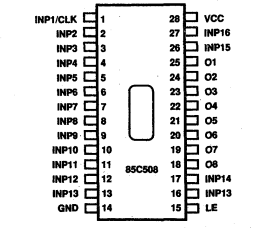


Figure 4. Intel 85C220 Serves as the Control Logic for DRAMS on the i860™ EV-AT



- TOTAL PROP. DELAY — • t_{PD} = 7.5ns
- LATCH ENABLE TO OUT — • t_{CO} = 4.5ns
- SETUP TO LATCH FREQUENCY — • t_{SU} = 5.5ns
- POWER SUPPLY CURRENT (130MHz) — • ICC = 48mA

Figure 5. 85C508 Summary

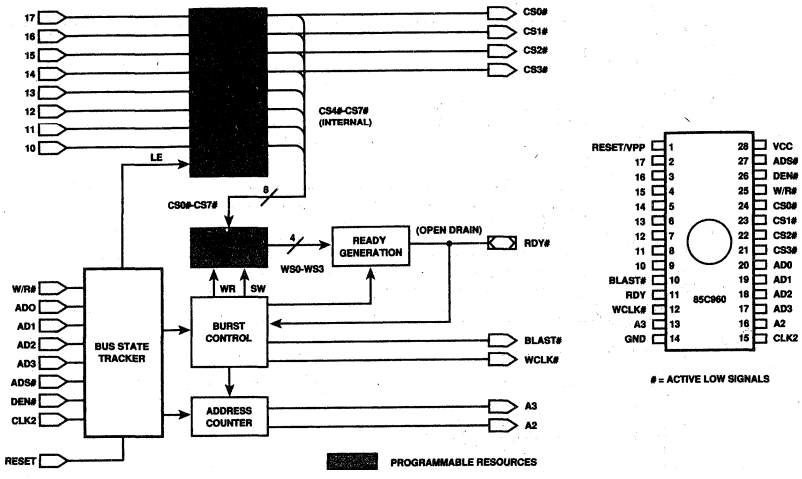


Figure 6. 85C960 PLD Pinout and Block Diagram

A 15 ns 2500 Gate Highly Flexible CHMOS EPLD

by Ronald W. Swartz and Michael J. Allen

ABSTRACT

A 2500 gate, 15 ns CMOS EPLD with configurable inputs, expandable sum of products (SOP), and SOP implementation of control signals has been developed. Independent synchronous or asynchronous clocking of the inputs and outputs, plus both internal and pad feedback of each macrocell, make this an extremely configurable 40-pin logic device.

INTRODUCTION

As system complexity and performance increase, system designers are requiring programmable logic devices that offer increased density, performance, and flexibility. One method of increasing the flexibility is to vary the number of product terms assigned per output [1]. The drawbacks of this approach are that it limits performance of the large macrocells, it is inefficient in silicon area and internal chip resources, and it also restricts the options for the user pinout. By developing a 40-pin CMOS EPLD which offers configurable input structures, expandable SOP macrocells, SOP internal control signals, configurable output structures, and dual feedback paths, the demand for a high density, high performance, highly flexible device is met. Typical Tpd is 15 ns and Tco is 12 ns. The 1μ N-well CMOS EPROM based technology allows for reconfigurability, low power operation, and 100% factory testing.

CHIP ARCHITECTURE

The device consists of 24 I/Os, 10 Latched Inputs (LINPs), and 2 Clock/Inp pins, all of which are TTL compatible. In addition to these signals, internal feedback from each macrocell is also available as an array input, allowing the I/O pad to be used as an additional device input. Each I/O and LIN is configured on power-up by clocking the programmed values of EPROM cells through a shift register. There are 17 product terms (bit lines) per macrocell. These consist of 2 groups of 4 product terms for sum of product output, 4 sets of 2 product terms for sum of product control signals, and 1 bit line for the architecture bits of that macrocell. Twelve more product terms are required for asynchronous input clocking, input architectures, and the security bit (verify inhibit). Thus there are 120 word lines and 420 bit lines in the device. The 420 bit lines are divided into 4 quadrants with 6 macrocells each. The word lines are global to all 4 quadrants. Figure 1 shows a block diagram of the device.

The LINPs may be individually configured as either latches or registers, with either a synchronous (pin) or asynchronous (product term) clock (ILE). For flow through operation, the asynchronous ILE term is programmed to be always enabled. Figure 2 shows the LINP input structure. The output macrocells may be configured to be combinatorial or registered, and support programmable output polarity. If the registered option is selected, further options on flip-flop type (D, T, JK, RS) and clocking (synchronous or asynchronous) may be selected. Each macrocell can

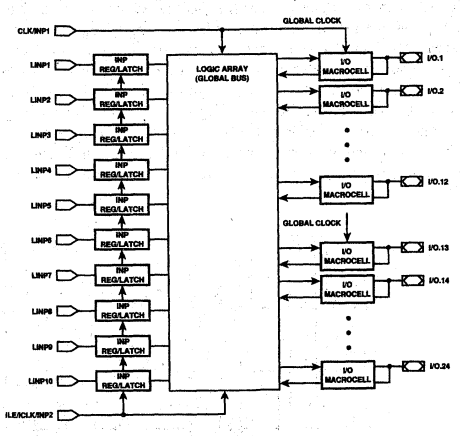


Figure 1. Block Diagram

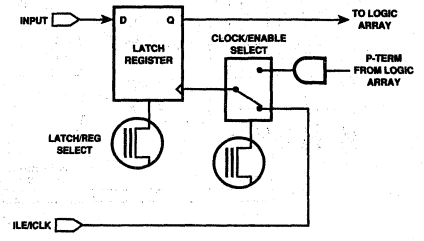


Figure 2. LINP Input Structure

allocate, in groups of 4, product terms to each of its neighbors. Thus the number of product terms summed into a given output can range from 0 to 16. A macrocell with 0 data product terms retains its control product terms, and can still be used as an SR or T flip flop. Two product terms are summed together for each of the macrocell's control signals; OE, SET, RESET, and ASYNCH CLOCK. Figure 3 shows the output macrocell architecture.

CIRCUITS AND PERFORMANCE

The high speed performance of the device may be attributed to a 1μ poly-silicide process, layout architecture, and circuit techniques. The process particulars will be described later in the paper, but were frozen for this design. Minimizing the parasitic delays was a high priority, and influenced several early layout choices. The large number of I/O macrocells presented a potential noise problem during multiple switching. To alleviate this, the device makes use of 2 Vcc and 2 VSS pins, which are located close to the center of the package, as shown in the pinout of Figure 4. This location has reduced pin inductance compared to the traditional corner placement, and significantly reduces power supply noise during simultaneous transitions.

The array of the device was broken into 4 quadrants of 105 columns each to reduce the rowline RC delays to less than 1 ns. All 4 quadrants are driven centrally from a single large row driver located at the end of the macrocell. By locating the row drivers here, the severe pitch limitations encountered by placing the row drivers on the end of the array are eliminated, and larger, faster tri-stateable drivers can be used. These drivers use high voltage p-channel devices to isolate the row drivers from the row lines during programming, with minimal speed degradation during normal mode. The bit line sense-amp, shown in Figure 5, contains

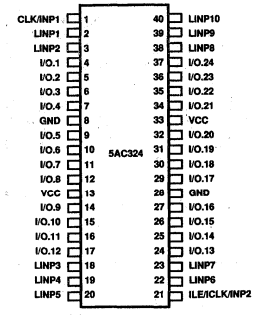


Figure 4. Device Pinout

unique biasing circuitry which limits the bit line swing to less than 150 mV, even when multiple cells are conducting. The amplifier uses a low threshold enhancement device to provide a pullup current, and a dynamic reference bias along with a current limiting device to limit the bit line swing. Response is very fast, even for recovery from super zero conditions, and typically takes less than 3 ns from row line input to sense amp output, allowing a Tpd of less than 15 ns. Figure 6 shows an oscillograph of typical Tpd data. Active Icc is less than 150 mA at 40 MHz. An option to trade off speed for power savings is also available to the user, which allows the part to operate in battery powered applications [2].

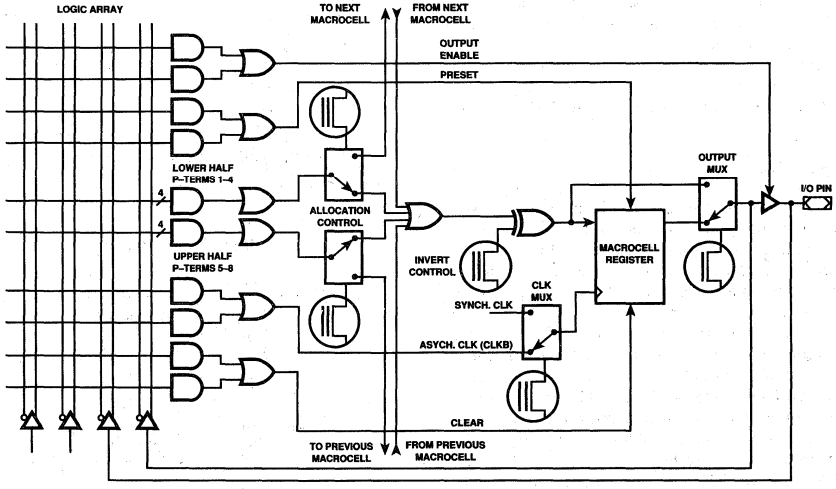


Figure 3. Output Macrocell Architecture

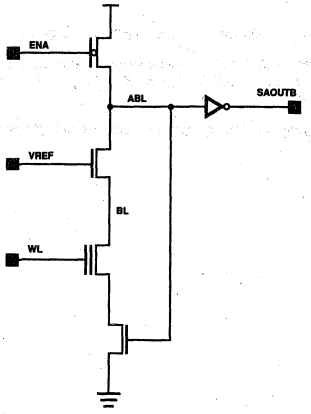


Figure 5. Bit Line Sense Amp

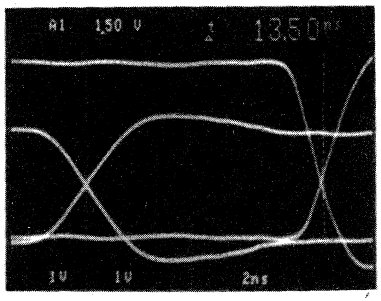


Figure 6. Tpd

Table 1 shows typical device performance.

Table 1. Typical Device Characteristics

Die Size	4.5 mm × 5.9 mm
Cell Size	4.2μ × 9.0μ
Tpd	15 ns
Tco	12 ns
Active Icc (40 MHz)	150 mA
Standby Icc	80 μA
Package	40-pin dip 44-pin PLCC
Technology	N-well 1.0μ CMOS EPROM with silicide

PROCESS PARAMETERS

The device has been fabricated in a 1.0μ N-well CMOS EPROM technology, and utilizes 2 polysilicon and 1 metalization layers. Tungsten silicide is used to reduce poly interconnect resistance. Table 2 summarizes Process Parameters. Programming circuitry is handled by high voltage p-channel devices. The cell is a special double poly FLOTOX structure that has been optimized for PLD applications.

Table 2. Process Parameters

Technology	1.0μ CMOS EPROM Double poly with silicide
N-channel Leff	.9μ
P-channel Leff	.9μ
Tox	250Å
Polycide resistivity	4 ohms/sq
Polysilicon pitch	2.5μ
Metal-1 pitch	2.8μ
Contact cut	1.2μ × 1.2μ

CONCLUSION

The combination of high speed, high density and flexible architecture makes this device an ideal solution for high speed microcomputer system design [3].

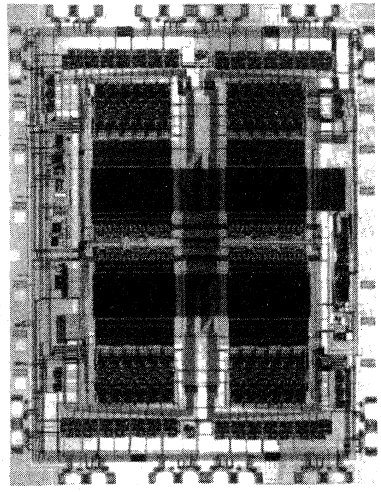


Figure 7. Die Photo

ACKNOWLEDGEMENTS

We would like to thank Jim Negrey and Reo Gargovich for their layout work, Chris Wawro, K.K. Ramakrishnan and Abid Asghar for their many helpful suggestions, and the Intel Fab personnel for producing the silicon.

REFERENCES

[1] Advanced Micro Devices, "Programmable Array Logic Handbook," 1986.

[2] S.C. Wong, et al, "CMOS Erasable Programmable Logic Device with Zero Standby Power," ISSCC Digest of Technical Papers, Feb. 1986, pp 242-243.

[3] Intel Corporation, "Microcomputer Programmable Logic Handbook," 1989.

October 1989

Implementing Cascaded Logic in EPLDs

THOM BOWNS
J. R. DONNELL
APPLICATIONS ENGINEER
PROGRAMMABLE LOGIC

Order Number: 292003-002

IMPLEMENTING CASCADED LOGIC IN THE 5C121

CONTENTS

PROBLEM

Designs that utilize numerous levels of cascaded logic often result in excessive product terms when expressed in the sum-of-products form. Although this poses no problem when designing with discrete logic, EPLDs are generally optimized for the sum-of-product form. This stems from the architecture of the basic Macrocell.

Macrocells typically consist of a programmable AND array feeding a fixed width OR gate. Most Intel EPLDs (with a few exceptions) have a fixed OR gate width of eight product terms. For most applications, eight available product terms are sufficient. However certain designs require logic to be cascaded, which usually causes product term requirements to expand geometrically. One example where product terms become an issue is cascaded exclusive-OR (XOR) circuits. Here the number of product terms increase by 2 to the nth power, where n equals the number of XOR gates. If the number of product terms exceeds eight, the equation may not fit in the EPLD macrocell.

SOLUTION

There is a simple solution to reduce the product term requirements when using cascading XOR (or other) logic. Figure 1 shows a circuit cascading five exclusive ORs. As designed, this circuit expands to 32 product terms when expressed in the minimized sum-of-products form. (This is assuming that signals A thru F are single product terms themselves.) Figure 4 shows the minimized logic equation file produced by Intel's Logic Optimizing Compiler (iLOC).

An easy solution to fitting this logic into an EPLD is to cascade three exclusive ORs together and then send the result through some type of combinatorial feedback primitive, such as a COIF (Combinatorial Output—Input Feedback) or NOCF (No Output—Combinatorial Feedback). This signal can now be cascaded through two more XOR's to get the five total. This circuit is shown in Figure 2. Figure 4 shows the logic equation file for this implementation. Note the reduction in product terms from Figure 3.

The only penalty in this method is the added delay needed for the feedback path. The worst case t_{pd} (input to output delay) for the circuit in Figure 2 would be twice the specified T_{pd} (input to output delay) for the target EPLD. For the 10 ns 85C220, the T_{pd} would be 20 ns worst case as implemented in Figure 3.

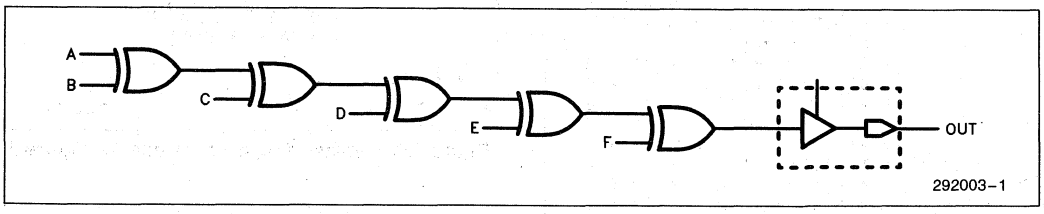


Figure 1. Cascaded Exclusive-ORs

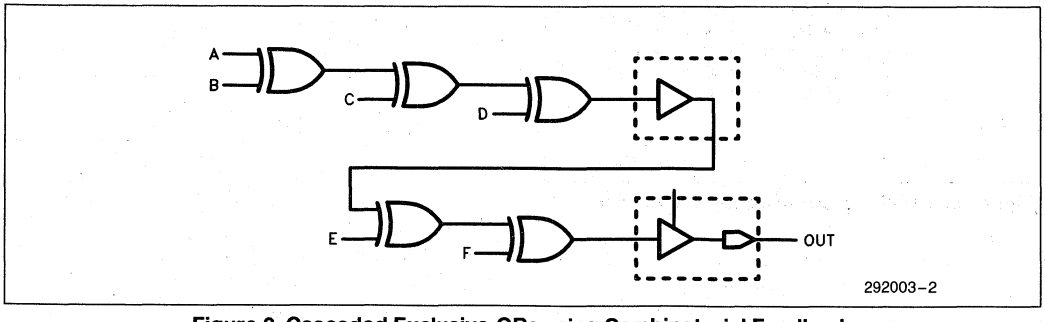


Figure 2. Cascaded Exclusive-ORs using Combinatorial Feedback

```
Cascading XOR gates--no feedback
LEF Version 2.0 Baseline 4.3i
22 Jul 1988

PART:   85C220

INPUTS: A, B, C, D, E, F
OUTPUTS: O

NETWORK:
        A = INP(A)
        B = INP(B)
        C = INP(C)
        D = INP(D)
        E = INP(E)
        F = INP(F)
        O = CONF(Oc, VCC)

EQUATIONS:
        Oc = F * E' * D' * C' * A' * B'
           + F' * E * D' * C' * A' * B'
           + F' * E' * D * C' * A' * B'
           + F * E * D * C' * A' * B'
           + F' * E' * D' * C * A' * B'
           + F * E * D' * C * A' * B'
           + F * E' * D * C * A' * B'
           + F' * E * D * C * A' * B'
           + F' * E' * D' * C' * A * B'
           + F * E * D' * C' * A * B'
           + F * E' * D * C' * A * B'
           + F' * E * D * C' * A * B'
           + F * E' * D' * C * A * B'
           + F' * E * D' * C * A * B'
           + F' * E' * D * C * A * B'
           + F * E * D * C * A * B'
           + F' * E' * D' * C' * A' * B
           + F * E * D' * C' * A' * B
           + F * E' * D * C' * A' * B
           + F' * E * D * C' * A' * B
           + F * E' * D' * C * A' * B
           + F' * E * D' * C * A' * B
           + F' * E' * D * C * A' * B
           + F * E * D * C * A' * B
           + F * E' * D' * C' * A * B
           + F' * E * D' * C' * A * B
           + F' * E' * D * C' * A * B
           + F * E * D * C' * A * B
           + F' * E' * D' * C * A * B
           + F * E * D' * C * A * B
           + F * E' * D * C * A * B
           + F' * E * D * C * A * B;

END$
```

292003-5

Figure 3. Minimized Logic Equations for Figure 1

```
Cascading XOR gates--
combinatorial feedback used
LEF Version 2.0 Baseline 4.3i
22 Jul 1988

PART:   85C220

INPUTS: A, B, C, D, E, F
OUTPUTS: O, NODE

NETWORK:
        A = INP(A)
        B = INP(B)
        C = INP(C)
        D = INP(D)
        E = INP(E)
        F = INP(F)
        O = CONF(Oc, VCC)
        NODE, NODE = COIF(NODEc, VCC)

EQUATIONS:
        NODEc = D * C' * A' * B'
              + D' * C * A' * B'
              + D' * C' * A * B'
              + D * C * A * B'
              + D' * C' * A' * B
              + D * C * A' * B
              + D * C' * A * B
              + D' * C * A * B;

        Oc = F * E' * NODE'
           + F' * E * NODE'
           + F' * E' * NODE
           + F * E * NODE;

END$
```

292003-6

Figure 4. Minimized Logic Equations for Figure 3

February 1987

16-Bit Binary Counter Implementation Using the 5C060 EPLD

KARL-HEINZ WEIGL
INTEL CORPORATION
MUNICH, GERMANY

Order Number: 292015-002

16-BIT BINARY COUNTER IMPLEMENTATION USING THE 5C060 EPLD

CONTENTS

INTRODUCTION

System designers often use programmable logic devices to implement counters. Use of PLA devices lets the user build customized counters to suit individual applications. In most cases such counters are not available, 'off-the-shelf' SSI/MSI devices. In other applications, the PLA implementation allows the designer to squeeze the counter function along with other 'glue' tasks into a single PLA, with the attendant higher integration benefits.

Use of traditional 20-pin and 24-pin PLAs, however, does not allow for the construction of large counters having greater than 10 significant bits. This is because these traditional PLAs have register and product term restrictions (even the larger bipolar PLAs have only 8 to 10 registers and less than 8 product terms per register). In contrast, the 5C060 24-pin erasable programmable logic device (EPLD) contains 16 registers that are programmable as 'D', 'T', 'RS' or 'JK' types. These 16 programmable registers enable the construction of Up/Down counters with up to 16 significant bits.

This application brief details the implementation of a 16-bit binary counter in the 5C060 EPLD. The design also demonstrates efficient counter construction utilizing toggle flip-flops (T-FF) that allows for minimum product term utilization.

DESIGN OBJECTIVE

The objective of the design is to implement a counter with the following features: (i) 16-bit binary count, (ii) toggle flip-flops, (iii) asynchronous clear, (iv) RUN/$\overline{\text{STOP}}$ function and (v) UP/$\overline{\text{DOWN}}$ function. The function table is shown in Figure 1.

RESET	UP/DOWN	RUN/STOP	Function
X	X	0	Inhibit Counting
0	0	1	Count Down
0	1	1	Count Up
1	X	X	Reset All Outputs to 'LOW'

Figure 1

TOGGLE FLIP-FLOPS

Counters can be most effectively implemented in PLA architectures using toggle flip-flops. This is because counters constructed with 'D' type flip-flops require an additional product term for every successive significant bit, whereas toggle flip-flop implementation requires only **one** product term per significant bit. Thus, the toggle flip-flop counter design is more miserly in product term consumption than the 'D' register design. Since product term minimization is the key element to maximizing PLA utilization, the T-FF counter design is more efficient. The truth table for the toggle flip-flop is shown in Fig. 2.

T	Q(N)	Q (N + 1)
0	0	0
0	1	1
1	0	1
1	1	0

Figure 2

SOLUTION

The 16-bit binary counter function was implemented in the 5C060 EPLD using the Intel Programmable Logic Development System (iPLDS). The equations for the 16-bit binary counter with the RESET, UP/$\overline{\text{DOWN}}$ and RUN/$\overline{\text{STOP}}$ functions are shown in the 'EQUATIONS' section of the LEF (Fig. 4). The pinout of the 5C060 with the implemented counter is shown in the RPT file (Utilization Report) Fig. 5. This RPT file also shows, under the 'OUTPUTS' section, that in each macrocell only one out of 8 product terms is used. In contrast the same 16-bit counter designed using 'D' type flip-flops would have required more than 16 product terms for the last significant bit.

```
INTEL CORPORATION
JAN. 15, 1987
1
1.0
5C060
BINARY 16-BIT UP/DOWN COUNTER WITH RUN/STOP AND ASYNCH. RESET USING T-FF

LB Version 4.01, Baseline 27.1 4/9/86
OPTIONS: TURBO=ON
PART: 5C060
INPUTS: RS,CLOCK,RESET,UD
OUTPUTS: Q0,Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8,Q9,QA,QB,QC,QD,QE,QF
NETWORK:
Q0,Q0F = TOTF (Q0T,CLK,CLR,GND,VCC)
Q1,Q1F = TOTF (Q1T,CLK,CLR,GND,VCC)
Q2,Q2F = TOTF (Q2T,CLK,CLR,GND,VCC)
Q3,Q3F = TOTF (Q3T,CLK,CLR,GND,VCC)
Q4,Q4F = TOTF (Q4T,CLK,CLR,GND,VCC)
Q5,Q5F = TOTF (Q5T,CLK,CLR,GND,VCC)
Q6,Q6F = TOTF (Q6T,CLK,CLR,GND,VCC)
Q7,Q7F = TOTF (Q7T,CLK,CLR,GND,VCC)
Q8,Q8F = TOTF (Q8T,CLK,CLR,GND,VCC)
Q9,Q9F = TOTF (Q9T,CLK,CLR,GND,VCC)
QA,QAF = TOTF (QAT,CLK,CLR,GND,VCC)
QB,QBF = TOTF (QBT,CLK,CLR,GND,VCC)
QC,QCF = TOTF (QCT,CLK,CLR,GND,VCC)
QD,QDF = TOTF (QDT,CLK,CLR,GND,VCC)
QE,QEF = TOTF (QET,CLK,CLR,GND,VCC)
QF = TONF (QFT,CLK,CLR,GND,VCC)
Q0T = OR (Q0U,Q0D)
CLK = INP (CLOCK)
CLR = INP (RESET)
Q1T = OR (Q1U,Q1D)
Q2T = OR (Q2U,Q2D)
Q3T = OR (Q3U,Q3D)
Q4T = OR (Q4U,Q4D)
Q5T = OR (Q5U,Q5D)
Q6T = OR (Q6U,Q6D)
Q7T = OR (Q7U,Q7D)
Q8T = OR (Q8U,Q8D)
Q9T = OR (Q9U,Q9D)
QAT = OR (QAU,QAD)
QBT = OR (QBU,QBD)
QCT = OR (QCU,QCD)
QDT = OR (QDU,QDD)
QET = OR (QEU,QED)
QFT = OR (QFU,QFD)
RS = INP (RS)
UD = INP (UD)
NUD = NOT (UD)
Q0U = AND (UD,RS)
```

292015-1

Figure 3. Example .ADF

```
Q1U = AND (UD,Q0F,Q0U)
Q2U = AND (UD,Q1F,Q1U)
Q3U = AND (UD,Q2F,Q2U)
Q4U = AND (UD,Q3F,Q3U)
Q5U = AND (UD,Q4F,Q4U)
Q6U = AND (UD,Q5F,Q5U)
Q7U = AND (UD,Q6F,Q6U)
Q8U = AND (UD,Q7F,Q7U)
Q9U = AND (UD,Q8F,Q8U)
QAU = AND (UD,Q9F,Q9U)
QBU = AND (UD,QAF,QAU)
QCU = AND (UD,QBF,QBU)
QDU = AND (UD,QCF,QCU)
QEU = AND (UD,QDF,QDU)
QFU = AND (UD,QEF,QEU)
NQ0F = NOT (Q0F)
NQ1F = NOT (Q1F)
NQ2F = NOT (Q2F)
NQ3F = NOT (Q3F)
NQ4F = NOT (Q4F)
NQ5F = NOT (Q5F)
NQ6F = NOT (Q6F)
NQ7F = NOT (Q7F)
NQ8F = NOT (Q8F)
NQ9F = NOT (Q9F)
NQAF = NOT (QAF)
NQBF = NOT (QBF)
NQCF = NOT (QCF)
NQDF = NOT (QDF)
NQEF = NOT (QEF)
Q0D = AND (NUD,RS)
Q1D = AND (NUD,NQ0F,Q0D)
Q2D = AND (NUD,NQ1F,Q1D)
Q3D = AND (NUD,NQ2F,Q2D)
Q4D = AND (NUD,NQ3F,Q3D)
Q5D = AND (NUD,NQ4F,Q4D)
Q6D = AND (NUD,NQ5F,Q5D)
Q7D = AND (NUD,NQ6F,Q6D)
Q8D = AND (NUD,NQ7F,Q7D)
Q9D = AND (NUD,NQ8F,Q8D)
QAD = AND (NUD,NQ9F,Q9D)
QBD = AND (NUD,NQAF,QAD)
QCD = AND (NUD,NQBF,QBD)
QDD = AND (NUD,NQCF,QCD)
QED = AND (NUD,NQDF,QDD)
QFD = AND (NUD,NQEF,QED)
END$
```

292015-2

Figure 3. Example .ADF (Continued)

```
INTEL CORPORATION
JAN. 15, 1987
1
1.0
5C060
BINARY 16-BIT UP/DOWN COUNTER WITH RUN/STOP AND ASYNCH. RESET USING T-FF

LB Version 4.01, Baseline 27.1 4/9/86
LEF Version 4.01 Baseline 22.2 2/4/86
OPTIONS: TURBO=ON
PART:
        5C060
INPUTS:
        RS, CLOCK, RESET, UD
OUTPUTS:
        Q0, Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q9, QA, QB, QC, QD, QE, QF
NETWORK:
        CLK = INP(CLOCK)
        RS = INP(RS)
        CLR = INP(RESET)
        UD = INP(UD)
        Q0, Q0F = TOTF(Q0T, CLK, CLR, GND, VCC)
        Q1, Q1F = TOTF(Q1T, CLK, CLR, GND, VCC)
        Q2, Q2F = TOTF(Q2T, CLK, CLR, GND, VCC)
        Q3, Q3F = TOTF(Q3T, CLK, CLR, GND, VCC)
        Q4, Q4F = TOTF(Q4T, CLK, CLR, GND, VCC)
        Q5, Q5F = TOTF(Q5T, CLK, CLR, GND, VCC)
        Q6, Q6F = TOTF(Q6T, CLK, CLR, GND, VCC)
        Q7, Q7F = TOTF(Q7T, CLK, CLR, GND, VCC)
        Q8, Q8F = TOTF(Q8T, CLK, CLR, GND, VCC)
        Q9, Q9F = TOTF(Q9T, CLK, CLR, GND, VCC)
        QA, QAF = TOTF(QAT, CLK, CLR, GND, VCC)
        QB, QBF = TOTF(QBT, CLK, CLR, GND, VCC)
        QC, QCF = TOTF(QCT, CLK, CLR, GND, VCC)
        QD, QDF = TOTF(QDT, CLK, CLR, GND, VCC)
        QE, QEF = TOTF(QET, CLK, CLR, GND, VCC)
        QF = TONF(QFT, CLK, CLR, GND, VCC)
EQUATIONS:
        QFT = UD' * QEF' * QDF' * QCF' * QBF' * QAF' * Q9F' * Q8F' * Q7F' * Q6F' *
              Q5F' * Q4F' * Q3F' * Q2F' * Q1F' * Q0F' * RS
            + UD * QEF * QDF * QCF * QBF * QAF * Q9F * Q8F * Q7F * Q6F * Q5F *
              Q4F * Q3F * Q2F * Q1F * Q0F * RS;

        QET = UD' * QDF' * QCF' * QBF' * QAF' * Q9F' * Q8F' * Q7F' * Q6F' * Q5F' *
              Q4F' * Q3F' * Q2F' * Q1F' * Q0F' * RS
            + UD * QDF * QCF * QBF * QAF * Q9F * Q8F * Q7F * Q6F * Q5F * Q4F *
              Q3F * Q2F * Q1F * Q0F * RS;

        QDT = UD' * QCF' * QBF' * QAF' * Q9F' * Q8F' * Q7F' * Q6F' * Q5F' * Q4F' *
              Q3F' * Q2F' * Q1F' * Q0F' * RS
            + UD * QCF * QBF * QAF * Q9F * Q8F * Q7F * Q6F * Q5F * Q4F * Q3F *
              Q2F * Q1F * Q0F * RS;
```

292015-3

Figure 4. Example .LEF

```
        QCT = UD' * QBF' * QAF' * Q9F' * Q8F' * Q7F' * Q6F' * Q5F' * Q4F' * Q3F' *
              Q2F' * Q1F' * Q0F' * RS
            + UD * QBF * QAF * Q9F * Q8F * Q7F * Q6F * Q5F * Q4F * Q3F * Q2F *
              Q1F * Q0F * RS;

        QBT = UD' * QAF' * Q9F' * Q8F' * Q7F' * Q6F' * Q5F' * Q4F' * Q3F' * Q2F' *
              Q1F' * Q0F' * RS
            + UD * QAF * Q9F * Q8F * Q7F * Q6F * Q5F * Q4F * Q3F * Q2F * Q1F *
              Q0F * RS;

        QAT = UD' * Q9F' * Q8F' * Q7F' * Q6F' * Q5F' * Q4F' * Q3F' * Q2F' * Q1F' *
              Q0F' * RS
            + UD * Q9F * Q8F * Q7F * Q6F * Q5F * Q4F * Q3F * Q2F * Q1F * Q0F *
              RS;

        Q9T = UD' * Q8F' * Q7F' * Q6F' * Q5F' * Q4F' * Q3F' * Q2F' * Q1F' * Q0F' *
              RS
            + UD * Q8F * Q7F * Q6F * Q5F * Q4F * Q3F * Q2F * Q1F * Q0F * RS;

        Q8T = UD' * Q7F' * Q6F' * Q5F' * Q4F' * Q3F' * Q2F' * Q1F' * Q0F' * RS
            + UD * Q7F * Q6F * Q5F * Q4F * Q3F * Q2F * Q1F * Q0F * RS;

        Q7T = UD' * Q6F' * Q5F' * Q4F' * Q3F' * Q2F' * Q1F' * Q0F' * RS
            + UD * Q6F * Q5F * Q4F * Q3F * Q2F * Q1F * Q0F * RS;

        Q6T = UD' * Q5F' * Q4F' * Q3F' * Q2F' * Q1F' * Q0F' * RS
            + UD * Q5F * Q4F * Q3F * Q2F * Q1F * Q0F * RS;

        Q5T = UD' * Q4F' * Q3F' * Q2F' * Q1F' * Q0F' * RS
            + UD * Q4F * Q3F * Q2F * Q1F * Q0F * RS;

        Q4T = UD' * Q3F' * Q2F' * Q1F' * Q0F' * RS
            + UD * Q3F * Q2F * Q1F * Q0F * RS;

        Q3T = UD' * Q2F' * Q1F' * Q0F' * RS
            + UD * Q2F * Q1F * Q0F * RS;

        Q2T = UD' * Q1F' * Q0F' * RS
            + UD * Q1F * Q0F * RS;

        Q1T = UD' * Q0F' * RS
            + UD * Q0F * RS;

        Q0T = RS;

END$
```

292015-4

Figure 4. Example .LEF (Continued)

```
Logic Optimizing Compiler Utilization Report
FIT Version 4.01 Baseline 27.1 4/9/86

 ***** Design implemented successfully

**** NOTE:  Connect signal CLOCK to  pin 1 AND pin 13.

INTEL CORPORATION
JAN. 15, 1987
1
1.0
5C060
BINARY 16-BIT UP/DOWN COUNTER WITH RUN/STOP AND ASYNCH. RESET USING T-FF

LB Version 4.01, Baseline 27.1 4/9/86
OPTIONS: TURBO=ON

                5C060
              - - - - -
      CLOCK -¦ 1   24¦- Vcc
        GND -¦ 2   23¦- RS
         Q7 -¦ 3   22¦- QF
         Q6 -¦ 4   21¦- QE
         Q5 -¦ 5   20¦- QD
         Q4 -¦ 6   19¦- QC
         Q3 -¦ 7   18¦- QB
         Q2 -¦ 8   17¦- QA
         Q1 -¦ 9   16¦- Q9
         Q0 -¦10   15¦- Q8
         UD -¦11   14¦- RESET
        GND -¦12   13¦- CLOCK
              - - - - -


**INPUTS**

                                                                  ¦          Feeds:
      Name  Pin  Resource   MCell #     PTerms                   ¦  MCells     OE   Clear  Clock

      CLOCK   1       INP         -          -                   ¦      -      -       -  CLK1
                                                                 ¦                        CLK2

         UD  11       INP         -          -                   ¦      1      -       -      -
                                                                 ¦      2
                                                                 ¦      3
                                                                 ¦      4
                                                                 ¦      5
                                                                 ¦      6
                                                                 ¦      7
                                                                 ¦      8
                                                                 ¦      9
                                                                 ¦     10
                                                                 ¦     11
                                                                 ¦     12
                                                                 ¦     13
                                                                 ¦     14
                                                                 ¦     15

        GND  12       GND         -          -                   ¦      -      -       -      -

      CLOCK  13       INP         -          -                   ¦      -      -       -  CLK1
                                                                 ¦                        CLK2

      RESET  14       INP         -          -                   ¦      -      -       1      -
                                                                 ¦                        2
                                                                 ¦                        3
                                                                 ¦                        4
                                                                 ¦                        5
                                                                 ¦                        6
                                                                 ¦                        7
                                                                 ¦                        8
                                                                 ¦                        9
                                                                 ¦                       10
                                                                 ¦                       11
                                                                 ¦                       12
                                                                 ¦                       13
                                                                 ¦                       14
                                                                 ¦                       15
                                                                 ¦                       16
```

292015-5

Figure 5. Example .RPT File

RS	23	INP	-	-	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16	-	-	-
Vcc	24	Vcc	-	-	-	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16	-	-

OUTPUTS

Name	Pin	Resource	MCell #	PTerms	MCells	Feeds: OE	Clear	Clock
Q7	3	TOTF	9	2/ 8	1 2 3 4 5 6 7 8	-	-	-
Q6	4	TOTF	10	2/ 8	1 2 3 4 5 6 7 8 9	-	-	-
Q5	5	TOTF	11	2/ 8	1 2 3 4 5 6 7 8 9 10	-	-	-
Q4	6	TOTF	12	2/ 8	1 2 3 4 5 6 7 8 9 10 11	-	-	-

292015-6

Figure 5. Example .RPT File (Continued)

Q3	7	TOTF	13	2/ 8	1 2 3 4 5 6 7 8 9 10 11 12	-	-	-
Q2	8	TOTF	14	2/ 8	1 2 3 4 5 6 7 8 9 10 11 12 13	-	-	-
Q1	9	TOTF	15	2/ 8	1 2 3 4 5 6 7 8 9 10 11 12 13 14	-	-	-
Q0	10	TOTF	16	1/ 8	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	-	-	-
Q8	15	TOTF	8	2/ 8	1 2 3 4 5 6 7	-	-	-
Q9	16	TOTF	7	2/ 8	1 2 3 4 5 6	-	-	-
QA	17	TOTF	6	2/ 8	1 2 3 4 5	-	-	-

292015-7

Figure 5. Example .RPT File (Continued)

```
        QB   18      TOTF        5      2/ 8        1          -     -     -
                                                     2
                                                     3
                                                     4

        QC   19      TOTF        4      2/ 8        1          -     -     -
                                                     2
                                                     3

        QD   20      TOTF        3      2/ 8        1          -     -     -
                                                     2

        QE   21      TOTF        2      2/ 8        1          -     -     -

        QF   22      TONF        1      2/ 8        -          -     -     -


**UNUSED RESOURCES**

      Name  Pin  Resource     MCell    PTerms

        -    2       -          -         -

**PART UTILIZATION**

95%     Pins
100%    MacroCells
24%     Pterms
```

292015-8

Figure 5. Example .RPT File (Continued)

intel®

APPLICATION BRIEF

AB-12

October 1989

Designing a Mailbox Memory for Two 80C31 Microcontrollers Using EPLDs

K. WEIGL & J. STAHL
INTEL CORPORATION
MUNICH, GERMANY

Order Number: 292016-004

DESIGNING A MAILBOX MEMORY FOR TWO 80C31 MICROCONTROLLERS USING EPLDs

CONTENTS

INTRODUCTION

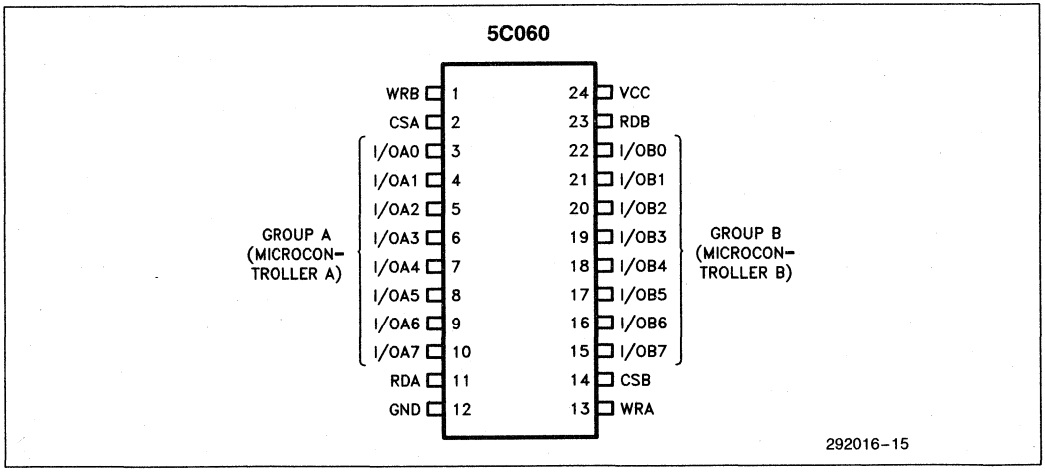
Very often, complex systems involve two or more microcontrollers to fulfill the requirements defined by a given objective. Since the nature of microcontrollers does not allow for easy dual-port memory design (no "READY" input; no "HOLD/HLDA" interface; port-oriented I/O etc.), design engineers are faced with the problem of interchanging information (data and status) between those microcontrollers. This application brief describes the design of a mailbox for exchanging information between two 80C31s, using a 5C060 EPLD as a "back-to-back" register, and a 5C032 EPLD as an arbitration vehicle to control the actions of the CPUs.

THE 5C060 MAILBOX

In this application, the 16 macrocells of the 5C060 are grouped into two sets of 8 so called "ROIF" (register output with input feedback) primitives to implement the two 8 bit bus interfaces needed. The grouping is done according to the following picture.

The 5C060 allows for independent clocking of 8 macrocells on each side of the chip, the two clock inputs are used to clock data from the microcontroller bus into the chip. To read the data written into the mailbox by one of the controllers, the RDA- (controller A is reading) or RDB- (controller B is reading) line must be pulled low by activating the read command (/RD). In order to avoid spurious read-cycles, the /RD commands from both microcontrollers are logically "ORed" together with an active high CS-signal (Chip Select) inside the 5C060. The CS-signal for both ports is derived from address line A15. Therefore, whenever A15 becomes a logic "1" (true), the mailbox is activated and ready to take or submit data.

Address range for the mailbox: F000 Hex to FFFF Hex
(Upper 12 kbyte)



THE 5C032 "MAILBOX CONTROLLER"

To keep the two microcontrollers informed about the status of their mailbox, the 5C032 is programmed to supply the following signals to both controllers:

/OBFA: "OUTPUT BUFFER FULL" FOR MC A

/OBFB: "OUTPUT BUFFER FULL" FOR MC B

/IBEA: "INPUT BUFFER EMPTY" FOR MC A

/IBEB: "INPUT BUFFER EMPTY" FOR MC B

/INTA: INTERRUPT TO MC A

/INTB: INTERRUPT TO MC B

The next section will discuss the meanings of these signals in more detail.

Output Buffer Full: This flag is set whenever the controller writes into its own output buffer. The flag remains valid, until the second controller has read the data. The flag is automatically reset to its inactive state when this read cycle is accomplished.

NOTE:
Both controllers can access (read or write) the mailbox simultaneously.

Input Buffer Empty: This flag indicates that there is no message in the mailbox. The flag will become inactive as soon as one microcontroller places a message for the other one (or vice versa).

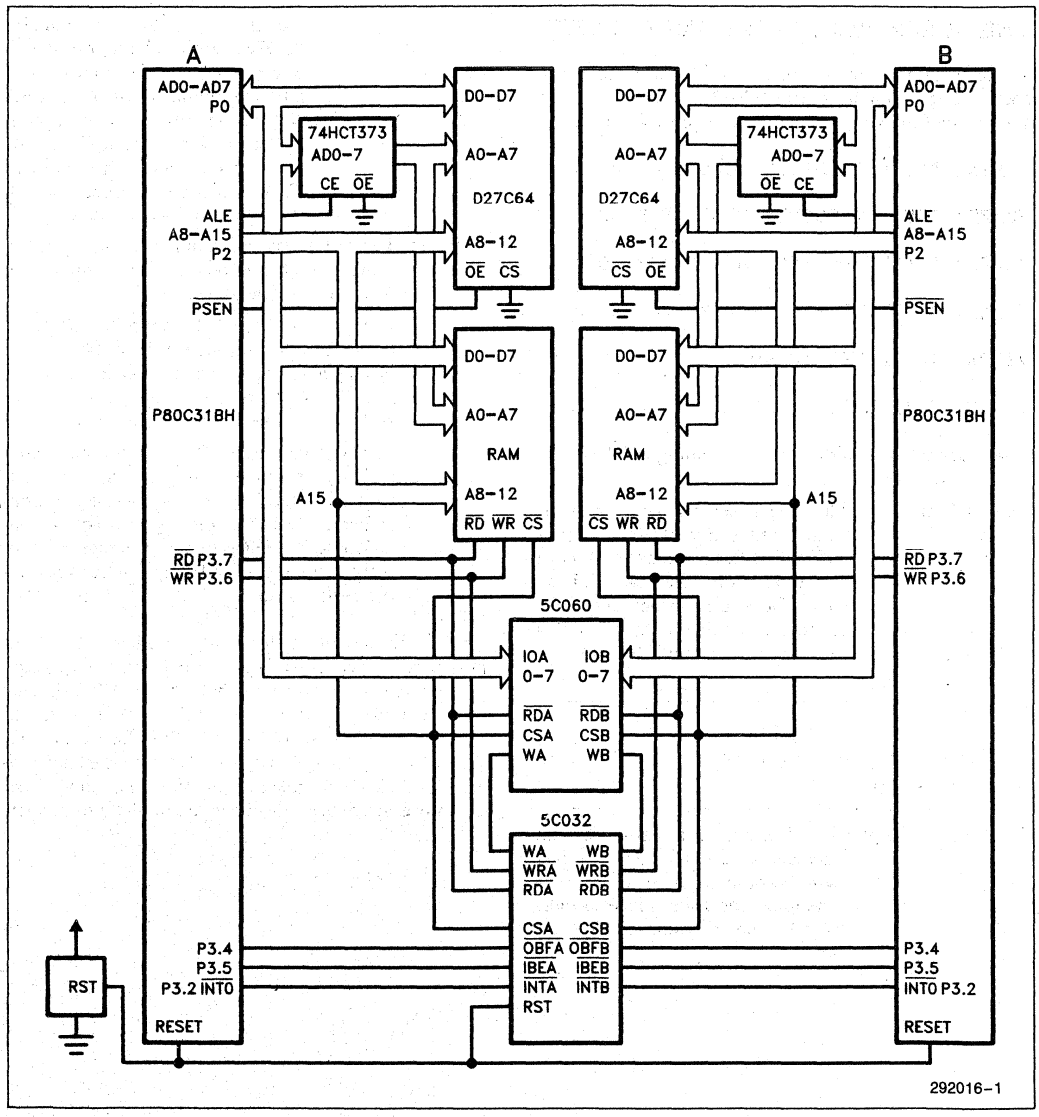
Example: /IBEA remains "LOW" until microcontroller B places a message for controller A into the mailbox for A. /IBEA will go "HIGH" as soon as controller B has accomplished its write cycle, and will not go "LOW" again until microcontroller A has read the message.

Interrupt: The 5C032 is programmed to supply interrupts to both microcontrollers involved, on one of the following events.

1. The /OBF flag of the opposite microcontroller becomes active; e.g. if controller A is placing a message for controller B, controller B receives an interrupt the same time as /OBFA becomes valid or vice versa.

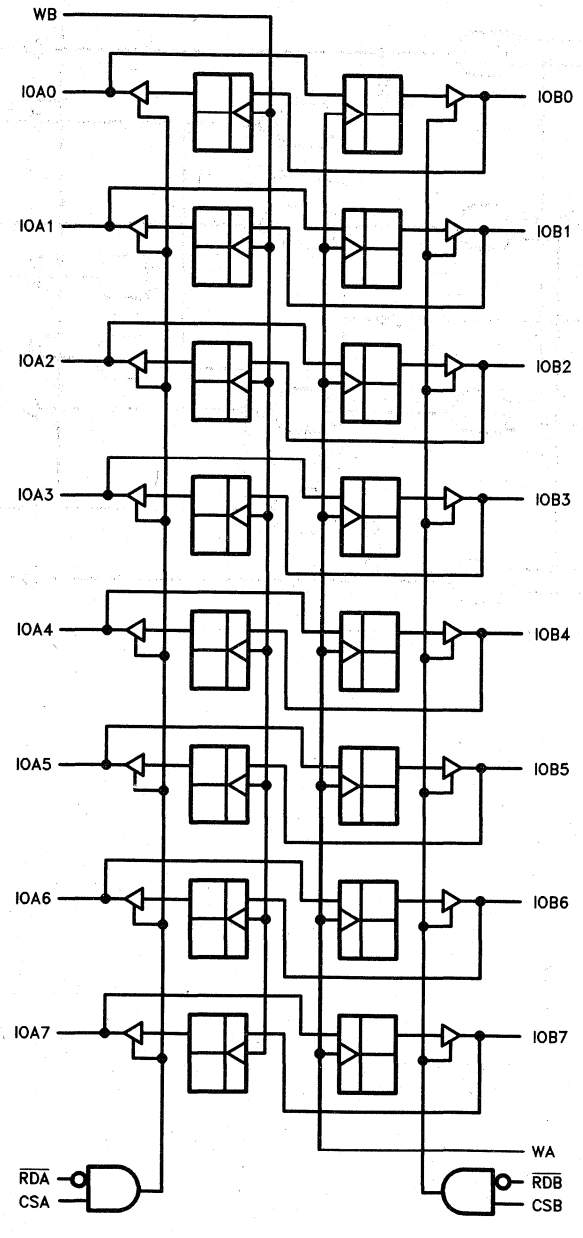
2. The /IBE flag of the opposite microcontroller goes active, indicating that this controller has received the message; e.g. if controller B reads the message stored by controller A, its /IBEB flag goes active and controller receives an interrupt indicating that the buffer is empty.

The signals described above are necessary to accomplish a secure handshake without overwriting messages accidentally. In addition to that, the 5C032 is issuing the actual write commands for the two register sets inside the 5C060. The /WRA and /WRB signals are results of logical "AND" functions between the appropriate CS- and /WR signals from the microcontrollers. Therefore, spurious write cycles are unlikely to happen.

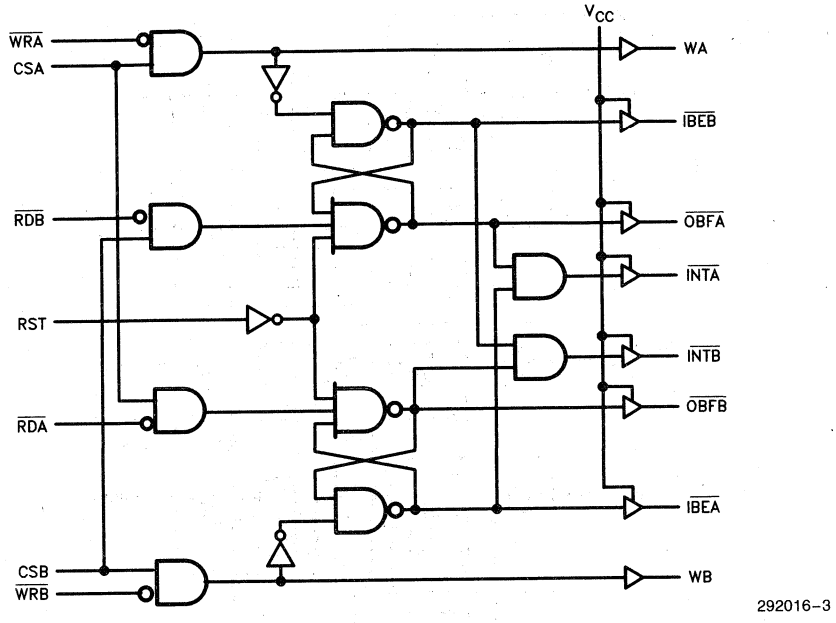


Block Diagram

5C060 "BACK TO BACK REGISTER"



5C032 "MAILBOX CONTROLLER"



292016-3

5C060 REGISTER ADF

```
JUERG STAHL
INTEL ZUERICH
August 28, 1989
80C31 MAILBOX MEMORY USING 5C060 / 5C032
REV 5
5C060

PART: 5C060
INPUTS: WB@1, CSA@2, CSB@14, nRDA@11, nRDB@23, WA@13
OUTPUTS: IOB7@15, IOA7@10, IOB6@16, IOA6@9,
         IOB5@17, IOA5@8, IOB4@18, IOA4@7,
         IOB3@19, IOA3@6, IOB2@20, IOA2@5,
         IOB1@21, IOA1@4, IOB0@22, IOA0@3

NETWORK:
        IOB7,DB7 = ROIF(DA7,WAC,GND,GND,RDBC)
        IOB6,DB6 = ROIF(DA6,WAC,GND,GND,RDBC)
        IOB5,DB5 = ROIF(DA5,WAC,GND,GND,RDBC)
        IOB4,DB4 = ROIF(DA4,WAC,GND,GND,RDBC)
        IOB3,DB3 = ROIF(DA3,WAC,GND,GND,RDBC)
        IOB2,DB2 = ROIF(DA2,WAC,GND,GND,RDBC)
        IOB1,DB1 = ROIF(DA1,WAC,GND,GND,RDBC)
        IOB0,DB0 = ROIF(DA0,WAC,GND,GND,RDBC)
        IOA7,DA7 = ROIF(DB7,WBC,GND,GND,RDAC)
        IOA6,DA6 = ROIF(DB6,WBC,GND,GND,RDAC)
        IOA5,DA5 = ROIF(DB5,WBC,GND,GND,RDAC)
        IOA4,DA4 = ROIF(DB4,WBC,GND,GND,RDAC)
        IOA3,DA3 = ROIF(DB3,WBC,GND,GND,RDAC)
        IOA2,DA2 = ROIF(DB2,WBC,GND,GND,RDAC)
        IOA1,DA1 = ROIF(DB1,WBC,GND,GND,RDAC)
        IOA0,DA0 = ROIF(DB0,WBC,GND,GND,RDAC)
        WAC = INP(WA)
        WBC = INP(WB)
        CSB = INP(CSB)
        CSA = INP(CSA)
        nRDB = INP(nRDB)
        nRDA = INP(nRDA)


EQUATIONS:

        RDBC = CSB * !nRDB;

        RDAC = CSA * !nRDA;



END$
```

292016-4

3

5C032 ARBITER ADF

```
JUERG STAHL
INTEL ZUERICH
August 28, 1989
80C31 MAILBOX MEMORY USING 5C060 / 5C032
REV 5
5C032

PART: 5C032
INPUTS: RST, nWRA, nRDB, CSA, nRDA, nWRB, CSB
OUTPUTS: WA, nOBFA, nIBEB, nINTA, nINTB, nOBFB, nIBEA, WB

NETWORK:
     nWRA = INP(nWRA)
     nRDA = INP(nRDA)
     CSA = INP(CSA)
     nWRB = INP(nWRB)
     nRDB = INP(nRDB)
     CSB = INP(CSB)
     RST = INP(RST)
     WA = CONF(WAd,VCC)
     WB = CONF(WBd,VCC)
     nOBFA,nOBFA = COIF(nOBFAd,VCC)
     nOBFB,nOBFB = COIF(nOBFBd,VCC)
     nIBEA,nIBEA = COIF(nIBEAd,VCC)
     nIBEB,nIBEB = COIF(nIBEBd,VCC)
     nINTA = CONF(nINTAd,VCC)
     nINTB = CONF(nINTBd,VCC)


EQUATIONS:

        nINTBd = nOBFB * nIBEB;
        nINTAd = nOBFA * nIBEA;
        nOBFBd = !(!(!nRDA * CSA) * nIBEA * !RST);
        nOBFAd = !(!(!nRDB * CSB) * nIBEB * !RST);
        nIBEBd = !(!(CSA * !nWRA) * nOBFA);
        nIBEAd = !(!(CSB * !nWRB) * nOBFB);
        WAd = CSA * !nWRA;
        WBd = CSB * !nWRB;


END$
```

292016-9

October 1988

3

Atypical Latch/Register Construction in EPLDs

THOM BOWNS
PROGRAMMABLE LOGIC APPLICATIONS
INTEL CORPORATION

Order Number: 292031-003

ATYPICAL LATCH/ REGISTER CONSTRUCTION IN EPLDs

CONTENTS

ATYPICAL LATCH/REGISTER CONSTRUCTION IN EPLDs

Though Intel's EPLDs include many of the typical latch and register types, some logic designs require register or latch configurations not directly supported in the current EPLDs. In many cases these register and latch configurations can be generated using the logic array and combinational feedback. A "latch" is defined as a level-triggered, flow-through type such as the 74373, and a "register" is defined as an edge-triggered flip-flop such as the 7474.

This application brief will detail the construction of a D-type latch, an RS latch and a D flip-flop using combinational logic and feedback. Also discussed is the construction of an RS flip-flop, a JK flip-flop and a T flip-flop using registered logic and feedback.

The RS latch is the simplest latch configuration. The equations for it are as follows: QB = !(Q + S), Q = !(QB + R) where Q is the output of one NOR gate, and QB is the output of the other (Note: as a convention in this Ap brief, the "!" operator is used to signify inversion). The schematic of the RS latch is shown in Figure 1a.

Since cross coupled logic is not supported in EPLDs, we must convert the equation to a single term with feedback.

```
QD, QF = COCF (Q, VCC)
Q = S + !R * QF;
```

where QF is the feedback from Q output.

This circuit can be implemented in an EPLD macrocell. Where combinational feedback is not supported, I/O feedback will suffice. The schematic of this implementation is shown in Figure 1b.

With the RS latch, the inputs are normally low. A logical one on S sets Q to 1, and a one on R resets Q to a 0. Logical ones on both inputs simultaneously cause the output to remain at a high level since S takes precedence over R in this implementation.

3

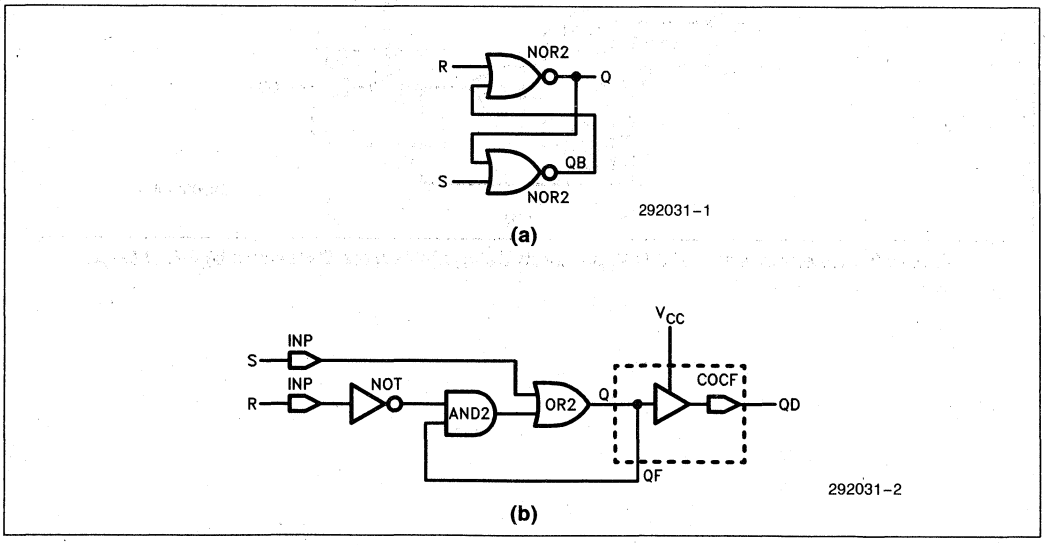


Figure 1. RS Latch Implementation In a) Discrete Gates and b) EPLD Logic

Another latch is the 74373 type, or D latch. This latch works by either enabling input data to appear at the output, or by holding the output to the last input data state. Its equation is this: QB = !(!(!D*E)*Q), Q = !(!(D*E)*QB). Again, Q is the output of one NAND gate, and QB is the output of the other. Figure 2a shows this version of the design.

Again, we must convert to an EPLD-type equation and schematic:

```
QD, QF = COCF (Q,VCC)
Q = D * E + !E * QF;
```

QF is the feedback from the COCF. In this circuit, when E is high, data flows through transparently. When E is brought low, data is latched. When using input feedback, care must be taken when tri-stating the output as data will no longer be latched. The EPLD implementation is given in Figure 2b.

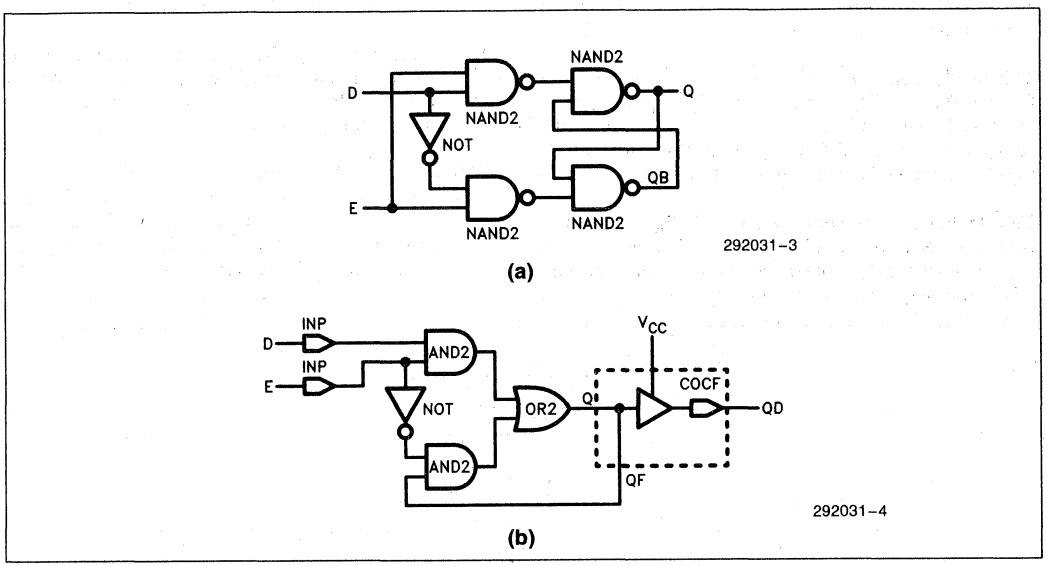


Figure 2. Implementation of a D Type Latch Using a) Discrete Gates and b) EPLD Logic

This latch can be cascaded with a second latch to produce an edge triggered, master/slave D flip-flop, using combinational logic. The flip-flop is a solution to using asynchronous clocking, preset and clear functions when they aren't supported. Also, if an I/O conflict exists within a macrocell group when using registered logic, this design will fit since it uses combinational logic. Figure 3 shows the schematic for this design.

This design does consume two macrocells, but in many cases, that isn't a problem.

The boolean equation of the D flip-flop is this:

```
QD,QF = COCF (Q,VCC)
YF = NOCF (Y)
Y = D * !CLOCK + YF * CLOCK;
Q = YF * CLOCK + QF * !CLOCK;
```

Q is the flip-flop output and Y is the first latch output. Data is latched in to the second latch on the low-going edge of clock, and is clocked out to Q on the high-going edge of clock.

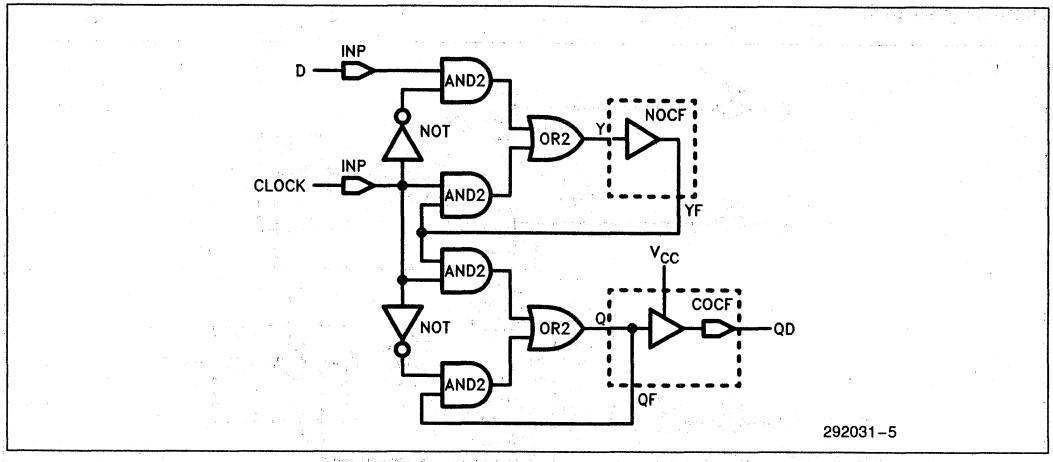


Figure 3. Combinational Logic Implementation of a D Flip-Flop

Preset and clear can be added into the equations as well:

```
QD,QF = COCF (Q,VCC)
YF = NOCF (Y)

Y = D * !CLOCK + YF * CLOCK;
Q = YF * CLOCK * ! (CLEAR TERM) +
    (PRESET TERM) +
    QF * !CLOCK * ! (CLEAR TERM);
```

When the PRESET TERM is logically true, Q is asynchronously set to 1.

When the CLEAR TERM is logically true, Q is asynchronously cleared to 0.
The PRESET TERM takes priority over the CLEAR TERM.
This schematic is shown in Figure 4.

Due to the nature of the design, input delays plus array delays plus feedback delays must be added and used to determine a maximum operating frequency. In this example, tIN + tAD + tCF + tAD = 113 ns for a −65 5C121, leaving a maximum frequency of 8.8 MHz.

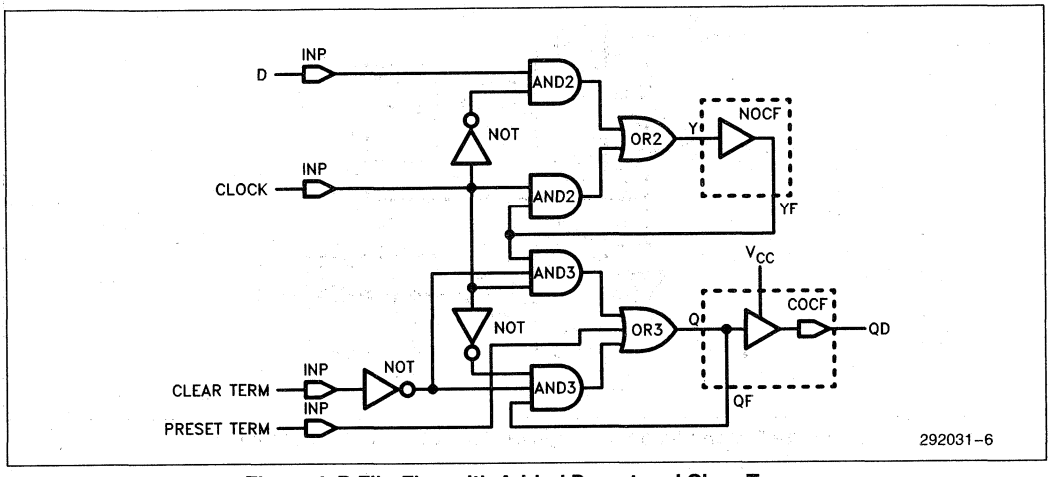


Figure 4. D Flip-Flop with Added Preset and Clear Terms

Other useful workarounds involve D registers and logic in constructing RS, JK and T flip-flops, for use in EPLDs not supporting these configurations. The RS flip-flop is simply the RS latch discussed earlier coupled to registered feedback.

```
QD,QF = RORF (Q,CLOCK,GND,GND,VCC)
Q = S + QF * !R;
```

Normally, S and R will remain low. When S is brought high, QD will become 1 on the next clock trigger edge. When R is brought high, QD will become 0 on the next clock trigger edge. The schematic is given in Figure 5.

The JK flip-flop is another useful and easily implemented register:

```
QD,QF = RORF (Q,CLOCK,GND,GND,VCC)
Q = J * !QF + !K * QF
```

When J = K = 1, QD toggles to opposite state on next clock trigger. When J = K = 0, QD remains the same. When J does not equal K, QD will follow J on next clock trigger. The schematic is shown in Figure 6.

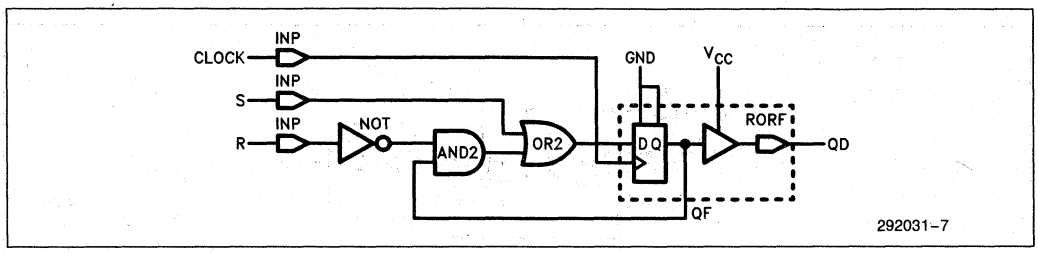


Figure 5. EPLD Implementation of an RS Flip-Flop

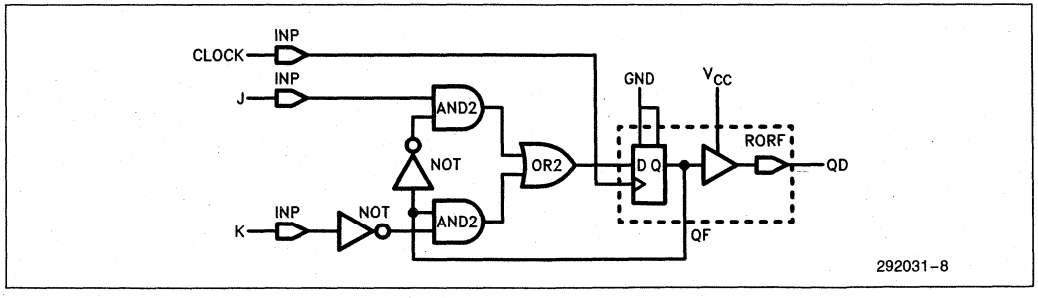


Figure 6. EPLD Implementation of a JK Flip-Flop

The T flip-flop is also easily constructed:

```
QD,QF = RORF (Q,CLOCK,GND,GND,VCC)
Q = T * ! QF + ! T * QF;
```

When T is high, QD will toggle to opposite state on next trigger. When T is low, QD will remain the same. Figure 7 shows the T flip-flop design schematic.

Each of these designs uses a minimum number of p-terms; adding p-terms is possible to the limit of the macrocell being used. It is possible to substitute an entire logical expression for each input listed (except register clock), as long as the minimized logic equations resulting do not exceed the macrocells p-term count.

For example, consider using the J-K register. Setting J = A * B * C + D and setting K = E * !F * !G + H + I then the minimized p-term count will expand from two p-terms to five p-terms, which would still be okay within a macrocell with more than five p-terms.

Using logic gates and combinational or registered feedback, one can easily implement many types of latches and registers. Regardless of the EPLD type, there exists the resources to implement any of the discussed circuitry.

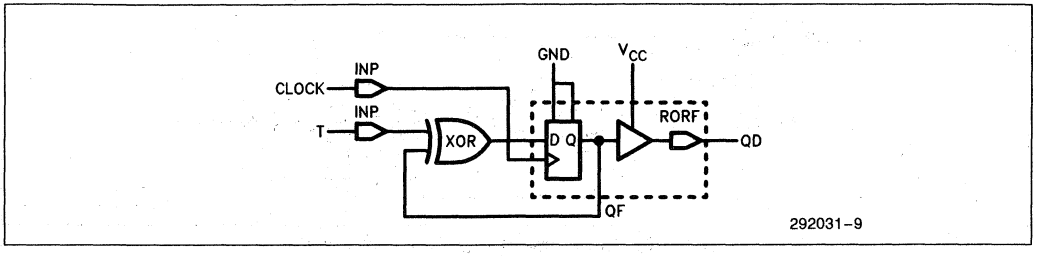


Figure 7. Implementation of a T Flip-Flop

June 1986

The 5C060 Unification of a CHMOS System

J. R. DONNELL
PROGRAMMABLE LOGIC APPLICATIONS
INTEL CORPORATION

Order Number: 292009-003

THE 5C060 UNIFICATION OF A CHMOS SYSTEM

CONTENTS

INTRODUCTION

From an outside glance, the world of computers and microprocessors seems filled with dedicated ICs that fulfill a variety of system needs. Upon closer inspection we find that designers must still reach into their bag of random logic to link together all of the parts of the system. It seems a shame to stuff a board full of high powered peripherals and still have portions of that board wasted on decoders, latches, and other miscellaneous random logic.

True, programmable logic has been around a long time. But that logic is somewhat rigid in form, one time programmable, and can also double as space heaters. These devices are totally unacceptable for a CMOS system. What is needed is a flexible PLA architecture, erasability for prototyping, and CMOS for low power. In addition, for this particular application the device must perform from static operation to 10 MHz.

OBJECTIVE

This application note covers the design of three separate circuits for Intel's CHMOS Design Kit. The functions performed by the 5C060 are: Memory decoding, wait state generation, and the power down circuitry for the 80C88 system clock.

MEMORY DECODING

The system in question supports one 32K bank of EPROM memory, and four banks of 4K static RAM. Figure 1 shows the memory map of this system. Address lines A19, A13, and A12 will be used to decode the address space. PWR_DWN and S2_MIO serve as enables. In addition, to avoid data bus contention signals memory read (MRDC) and advanced memory write (AMWC) are decoded along with the address lines for RAM chip selects. This is necessary for devices without output enables (OE) on multiplexed address/data busses.

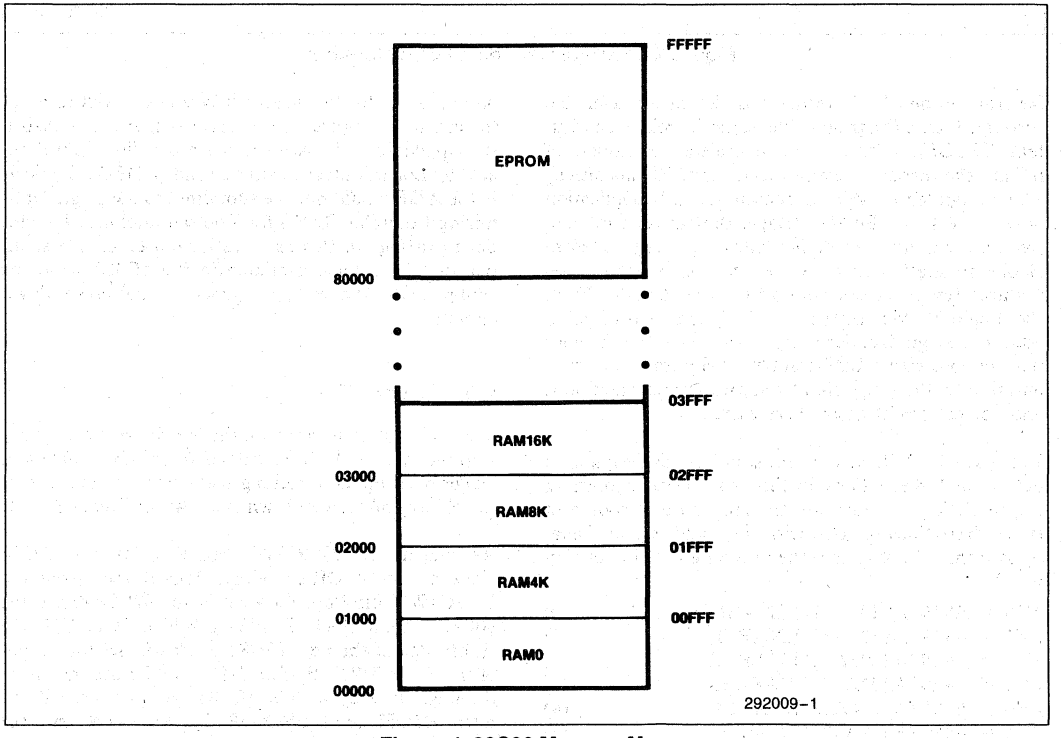


Figure 1. 80C88 Memory Map

Figure 2 shows a discrete implementation of the chip select decoding logic.

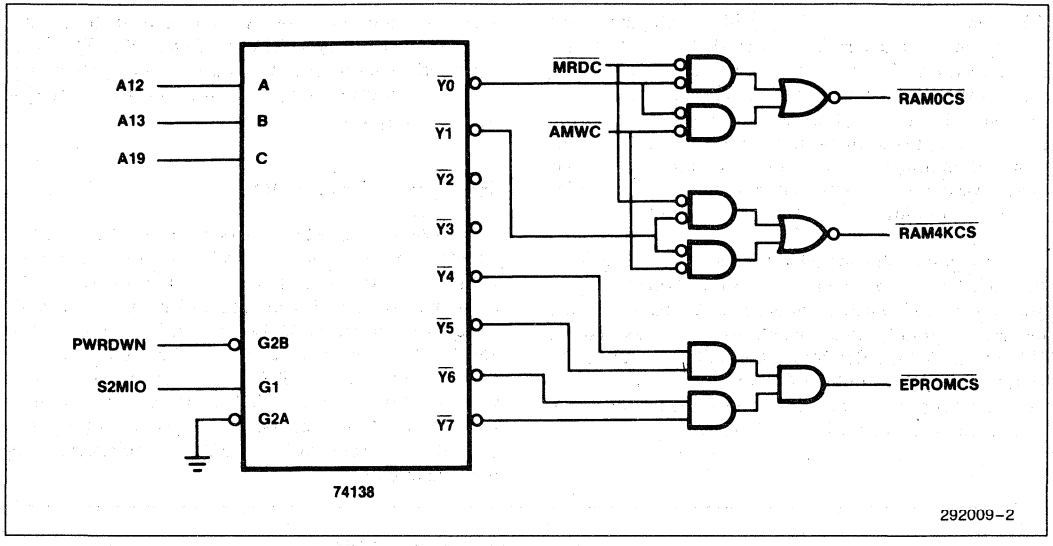


Figure 2. Discrete Decoding Logic Solution

Several options for entering this design are available through Intel's Programmable Logic Development System (iPLDS). (For a more complete description of iPLDS the reader is referred to the iPLDS data sheet.) The design entry vehicle chosen for this application note is the Logic Builder. (Logic Builder is an interactive netlist method of design entry especially suited to Boolean equation entry and entry from existing schematics.) Several reasons are behind this decision. First, the Logic Builder software is included in iPLDS. In addition, Logic Builder entry is very fast, the designer may choose either netlist entry or Boolean equations, and finally, the Logic Builder software makes additions and corrections of design very easy.

Using Logic Builder, the first step for this design is to determine the equations for the 3 to 8 decoder shown in Figure 2. These equations are simply the decoding of the address lines ANDed with the enable signal. Equations 0 thru 8 implement the decoding function of Figure 2.

```
/Y0 = /A19*/A13*/A12*ENABLE;          (0)
/Y1 = /A19*/A13*A12*ENABLE;           (1)
/Y2 = /A19*A13*/A12*ENABLE;           (2)
/Y3 = /A19*A13*A12*ENABLE;            (3)
/Y4 = A19*/A13*/A12*ENABLE;           (4)
/Y5 = A19*/A13*A12*ENABLE;            (5)
/Y6 = A19*A13*/A12*ENABLE;            (6)
/Y7 = A19*A13*A12*ENABLE;             (7)
ENABLE = /PWRDWN*S2MIO;               (8)
```

Armed with this knowledge it becomes trivial to enter the circuit of Figure 2 into Logic Builder. Included in the Appendix is the Advanced Design File (ADF) created by Logic Builder for this circuit (ADF-1). Typically the ADF would now be submitted to the Logic Optimizing Compiler (LOC) for Boolean minimization and design fitting. In this case we have used only a small portion if the logic available in the 5C060 so let us continue with the wait state generator and power down circuitry.

Power Down

Since this design is based on the 80C88 we can actually stop the system clock for extended periods of time and power back up as if nothing had occurred. The circuit to achieve this power down is shown in Figure 3.

As long as the PWRDWN signal is low the 82C84 clock output is OR'ed with a logical zero from the PWRDWN flip-flop. As a result the 82C84 drives the 80C88 system clock. If PWRDWN goes HIGH, the rising edge of the next 82C84 clock will set the output of the PWRDWN flip-flop HIGH inhibiting the fall of the next clock cycle. The 80C88 system clock will remain HIGH until PWRDWN goes LOW and the PWRDWN flip-flop is clocked from the 82C84 clock. Using this configuration we avoid partial clock cycles for the 80C88 system clock.

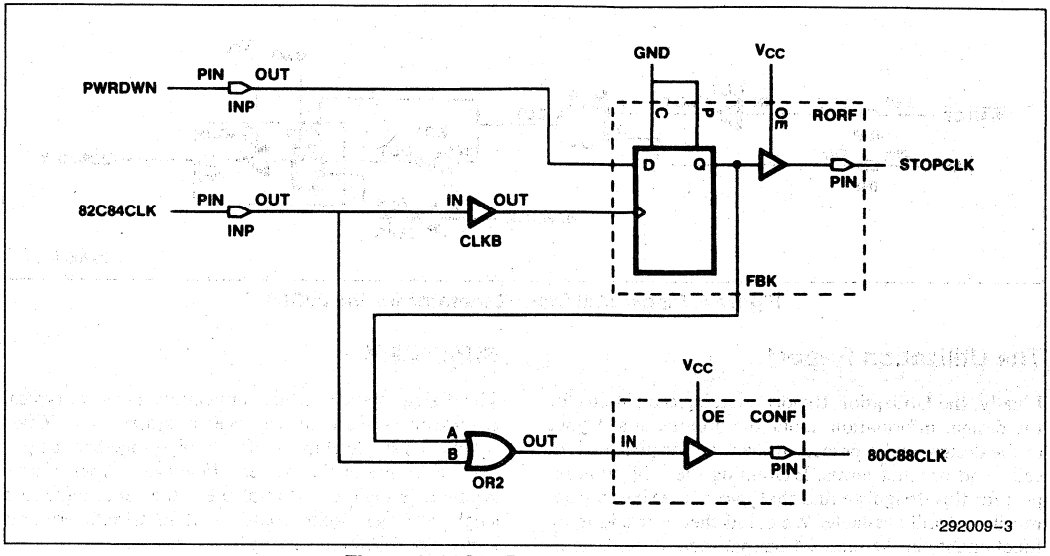


Figure 3. 80C88 Power Down Circuit

Again, entering this circuit into Logic Builder is trivial. In fact it can be added directly to the decoder circuit shown above. The ADF file for this addition is shown in the appendix under ADF-2.

Wait States

The majority of memory and peripheral devices which fail to operate at the maximum CPU frequency typically do not require more than one wait state. The circuit shown in Figure 4 is an example of a simple wait state generator. The circuit operation is as follows. Given that a memory location requiring a wait state has been selected, ALE in conjunction with /WAITCS will clear the flip-flop—driving the 82C84RDY line high low. The 82C84 samples the RDY line during T2 of the 80C88 bus cycle, and in this case detects a wait state. The rising edge of T2 then clocks the 82C84RDY line high thereby inserting only one wait state.

Once again, adding this circuit to the existing decoder and power down design is simple. The final ADF file is given in the appendix under ADF-3. Once the final design has been completed the ADF is submitted to the Logic Optimizing Compiler. LOC compiles the design, performs Boolean minimization, and fits the design into the target EPLD. In addition, LOC produces two files. The JEDEC programming file, the Logic Equation File (LEF), and the Utilization Report. These are also included in the appendix for each step in this design process.

LOC FILES

The JEDEC File

The JEDEC file is analogous to the object code file that is used to program EPROMs. This file is used by the Logic Programming Software (LPS) to program Intel's EPLDs.

The LEF File

The LEF file is an optional file produced by the compiler. The LEF file contains the minimized Boolean equations which resulted from the original ADF. Some interesting points can be raised concerning the LEF file. Looking at LEF-3, first recall that the EPROM chip select was a function of A19, A13, A12, and the enable signals. It turns out that after minimization the EPROM chip select depends only on A19 and the enable signals (/PWRDWN and S2MIO). This is shown in the LEF file. One other point, the initial wait state circuitry employed a JK flip-flop. The compiler automatically minimized this circuit into a D-type flip-flop with feedback achieving the same functionality.

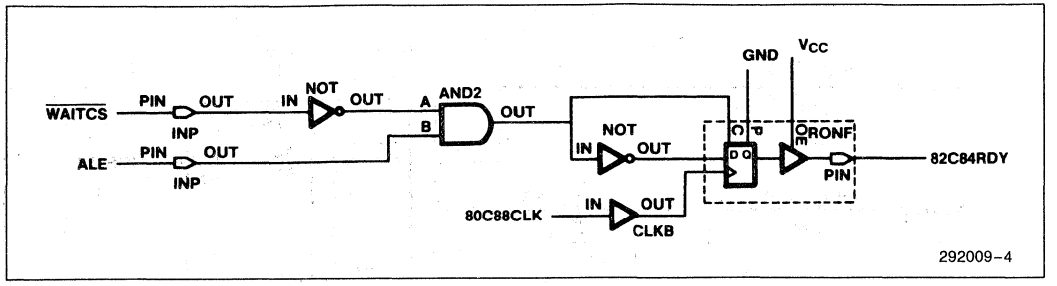


Figure 4. Single Wait State Generator for the 80C88

The Utilization Report

Finally, the Utilization Report contains the pin-out for the design, information about the architectural layout of the design, and a percent utilization for pins, macrocells, and product terms. Examining the utilization report for this design we find that two of the sixteen macrocells are still available. We could therefore add more functionality in the same 24 pin package. Possible additions would be more memory decoding, invalid memory detection, additional wait state generators, etc. One point should be raised: The circuitry designed in this applications note is relatively simple compared to the complex logic functions that could be implemented in the 5C060.

SUMMARY

The designs shown in this applications note are typical requirements of any microprocessor system. The 5C060 provided a single chip solution to bind together the primary elements of that system. Few other types of programmable logic could implement the same logic in a single package. None could do it in CMOS erasable logic. The 5C060 has room for more.

APPENDIX

ADF-1

```
JR Donnell
Intel
January 31, 1986
5C060
0
5C060
Decoder for 80C88 system - 16K RAM and upper 512K EPROM
LB Version 3.0, Baseline 17x, 9/26/85
PART: 5C060
INPUTS: A19,A13,A12,PWRDWN,S2MIO,AMWC,MRDC
OUTPUTS: RAM0CS,RAM4KCS,RAM8KCS,RAM16KCS,EPROMCS
NETWORK:
RAM0CS = CONF (RAM0CS,VCC)
RAM4KCS = CONF (RAM4KCS,VCC)
RAM8KCS = CONF (RAM8KCS,VCC)
RAM16KCS = CONF (RAM16KCS,VCC)
EPROMCS = CONF (EPROMCS,VCC)
A19 = INP (A19)
A13 = INP (A13)
A12 = INP (A12)
PWRDWN = INP (PWRDWN)
S2MIO = INP (S2MIO)
MRDC = INP (MRDC)
AMWC = INP (AMWC)
EQUATIONS:
RAM8KCS = /(/MRDC*Y2
        + /AMWC*Y2);
RAM16KCS = /(/MRDC*Y3
        + /AMWC*Y3);
EPROMCS = /(/Y7
        + /Y6
        + /Y5
        + /Y4);
Y7 = /(A19*A13*A12*ENABLE);
Y6 = /(A19*A13*/A12*ENABLE);
Y5 = /(A19*/A13*A12*ENABLE);
Y4 = /(A19*/A13*/A12*ENABLE);
ENABLE = /PWRDWN*S2MIO;
Y3 = /(/A19*A13*A12*ENABLE);
Y2 = /(/A19*A13*/A12*ENABLE);
RAM4KCS = /(/MRDC*Y1
        + /AMWC*Y1);
Y1 = /(/A19*/A13*A12*ENABLE);
RAM0CS = /(/MRDC*Y0
        + /AMWC*Y0);
Y0 = /(/A19*/A13*/A12*ENABLE);
END$
```

292009-5

ADF-2

```
JR Donnell
Intel
January 31, 1986
5C060
0
5C060
Decoder for 80C88 system - 16K RAM and upper 512K EPROM
Plus power down circuit
LB Version 3.0, Baseline 17x, 9/26/85
PART: 5C060
INPUTS: A19,A13,A12,PWRDWN,S2MIO,AMWC,MRDC,82C84CLK
OUTPUTS: RAM0CS,RAM4KCS,RAM8KCS,RAM16KCS,EPROMCS,STOPCLK,80C88CLK
NETWORK:
RAM0CS = CONF (RAM0CS,VCC)
RAM4KCS = CONF (RAM4KCS,VCC)
RAM8KCS = CONF (RAM8KCS,VCC)
RAM16KCS = CONF (RAM16KCS,VCC)
EPROMCS = CONF (EPROMCS,VCC)
STOPCLK,STOPCLKF = RORF (PWRDWN,82C84CLKB,GND,GND,VCC)
80C88CLK = CONF (80C88CLK,VCC)
PWRDWN = INP (PWRDWN)
82C84CLKB = CLKB (82C84CLK)
80C88CLK = OR (STOPCLKF,82C84CLK)
82C84CLK = INP (82C84CLK)
A19 = INP (A19)
A13 = INP (A13)
A12 = INP (A12)
S2MIO = INP (S2MIO)
MRDC = INP (MRDC)
AMWC = INP (AMWC)
EQUATIONS:
RAM0CS = /(/MRDC*Y0
         + /AMWC*Y0);
RAM4KCS = /(/MRDC*Y1
         + /AMWC*Y1);
RAM8KCS = /(/MRDC*Y2
         + /AMWC*Y2);
RAM16KCS = /(/MRDC*Y3
         + /AMWC*Y3);
EPROMCS = /(/Y7
         + /Y6
         + /Y5
         + /Y4);
Y0 = /(/A19*/A13*/A12*ENABLE);
Y1 = /(/A19*/A13*A12*ENABLE);
Y2 = /(/A19*A13*/A12*ENABLE);
Y3 = /(/A19*A13*A12*ENABLE);
Y7 = /(A19*A13*A12*ENABLE);
Y6 = /(A19*A13*/A12*ENABLE);
Y5 = /(A19*/A13*A12*ENABLE);
Y4 = /(A19*/A13*/A12*ENABLE);
ENABLE = /PWRDWN*S2MIO;
END$
```

292009-6

ADF-3

```
JR Donnell
Intel
January 31, 1986
5C060
0
5C060
Decoder for 80C88 system - 16K RAM and upper 512K EPROM
Plus power down circuit
Plus wait state circuit
LB Version 3.0, Baseline 17x, 9/26/85
PART: 5C060
INPUTS: A19,A13,A12,PWRDWN,S2MIO,AMWC,MRDC,82C84CLK,ALE,WAITCS
OUTPUTS: RAM0CS,RAM4KCS,RAM8KCS,RAM16KCS,EPROMCS,STOPCLK,80C88CLK,82C84RDY
NETWORK:
RAM0CS = CONF (RAM0CS,VCC)
RAM4KCS = CONF (RAM4KCS,VCC)
RAM8KCS = CONF (RAM8KCS,VCC)
RAM16KCS = CONF (RAM16KCS,VCC)
EPROMCS = CONF (EPROMCS,VCC)
STOPCLK,STOPCLKF = RORF (PWRDWN,82C84CLKB,GND,GND,VCC)
80C88CLK,80C88CLKF = COIF (80C88CLK,VCC)
82C84RDY = RONF (82C84RDYD,80C88CLKB,82C84RDYC,GND,VCC)
PWRDWN = INP (PWRDWN)
82C84CLKB = CLKB (82C84CLK)
80C88CLK = OR (STOPCLKF,82C84CLK)
82C84CLK = INP (82C84CLK)
A19 = INP (A19)
A13 = INP (A13)
A12 = INP (A12)
S2MIO = INP (S2MIO)
MRDC = INP (MRDC)
AMWC = INP (AMWC)
80C88CLKB = CLKB (80C88CLKF)
WAITCS = INP (WAITCS)
ALE = INP (ALE)
EQUATIONS:
RAM0CS = /(/MRDC*Y0
         + /AMWC*Y0);
RAM4KCS = /(/MRDC*Y1
         + /AMWC*Y1);
RAM8KCS = /(/MRDC*Y2
         + /AMWC*Y2);
RAM16KCS = /(/MRDC*Y3
         + /AMWC*Y3);
EPROMCS = /(/Y7
         + /Y6
         + /Y5
         + /Y4);
Y0 = /(/A19*/A13*/A12*ENABLE);
Y1 = /(/A19*/A13*A12*ENABLE);
Y2 = /(/A19*A13*/A12*ENABLE);
Y3 = /(/A19*A13*A12*ENABLE);
Y7 = /(A19*A13*A12*ENABLE);
Y6 = /(A19*A13*/A12*ENABLE);
Y5 = /(A19*/A13*A12*ENABLE);
Y4 = /(A19*/A13*/A12*ENABLE);
ENABLE = /PWRDWN*S2MIO;
82C84RDYD = /82C84RDYC;
82C84RDYC = /WAITCS*ALE;
END$
```

292009-7

LEF-3

```
JR Donnell
Intel
January 31, 1986
5C060
0
5C060
Decoder for 80C88 system - 16K RAM and upper 512K EPROM
Plus power down circuit
Plus wait state circuit
LB Version 3.0, Baseline 17x, 9/26/85
PART:
        5C060
INPUTS:
        A19, A13, A12, PWRDWN, S2MIO, AMWC, MRDC, 82C84CLK, ALE, WAITCS
OUTPUTS:
        RAM0CS, RAM4KCS, RAM8KCS, RAM16KCS, EPROMCS, STOPCLK, 80C88CLK,
        82C84RDY
NETWORK:
        A19 = INP(A19)
        A13 = INP(A13)
        A12 = INP(A12)
        PWRDWN = INP(PWRDWN)
        S2MIO = INP(S2MIO)
        AMWC = INP(AMWC)
        MRDC = INP(MRDC)
        82C84CLK = INP(82C84CLK)
        ALE = INP(ALE)
        WAITCS = INP(WAITCS)
        RAM0CS = CONF(RAM0CS, VCC)
        RAM4KCS = CONF(RAM4KCS, VCC)
        RAM8KCS = CONF(RAM8KCS, VCC)
        RAM16KCS = CONF(RAM16KCS, VCC)
        EPROMCS = CONF(EPROMCS, VCC)
        ..SG000D = CLKB(82C84CLKB)
        STOPCLK, STOPCLKF = RORF(PWRDWN, ..SG000D, GND, GND, VCC)
        80C88CLK, 80C88CLKF = COTF(80C88CLK, VCC)
        ..SG001D = CLKB(80C88CLKB)
        82C84RDY = RONF(82C84RDYD, ..SG001D, 82C84RDYC, GND, VCC)
EQUATIONS:
        82C84RDYC = WAITCS' * ALE;

        ..SG001D = 80C88CLKF;

        82C84RDYD = (WAITCS' * ALE)';

        80C88CLK = (STOPCLKF' * 82C84CLK')';

        ..SG000D = 82C84CLK;

        EPROMCS = (A19 * PWRDWN' * S2MIO)';

        RAM16KCS = MRDC * AMWC
                 + A19' * A13 * A12 * PWRDWN' * S2MIO;

        RAM8KCS = MRDC * AMWC
                + A19' * A13 * A12' * PWRDWN' * S2MIO;

        RAM4KCS = MRDC * AMWC
                + A19' * A13' * A12 * PWRDWN' * S2MIO;

        RAM0CS = MRDC * AMWC
               + A19' * A13' * A12' * PWRDWN' * S2MIO;

RND$
```

292009-8

RPT-3

```
Logic Optimizing Compiler Utilization Report

 ***** Design implemented successfully

JR Donnell
Intel
January 31, 1986
5C060
0
5C060
Decoder for 80C88 system - 16K RAM and upper 512K EPROM
Plus power down circuit
Plus wait state circuit
LR Version 3.0. Baseline 17x. 9/26/85

              5C060
            - - - - -
       GND -: 1   24:- Vcc
    PWRDWN -: 2   23:- A19
       GND -: 3   22:- STOPCLK
       GND -: 4   21:- R2C84RDY
    WAITCS -: 5   20:- 80C88CLK
       ALE -: 6   19:- EPROMCS
  R2C84CLK -: 7   18:- RAM16KCS
      MRDC -: 8   17:- RAM8KCS
      AMWC -: 9   16:- RAM4KCS
     S2MIO -:10   15:- RAM0CS
       A12 -:11   14:- A13
       GND -:12   13:- GND
            - - - - -

**INPUTS**

                                                                  Feeds:
      Name  Pin  Resource   MCell #   PTerms  :  MCells      OE   Clear  Clock

    PWRDWN    2       INP         -        -        1       -       -       -
                                                    4
                                                    5
                                                    6
                                                    7
                                                    8

    WAITCS    5       INP        11     0/ 8        2       -      -2       -

       ALE    6       INP        12     0/ 8        2       -       2       -

 R2C84CLK    7       INP        13     0/ 8        3       -       -       1

      MRDC    8       INP        14     0/ 8        5       -       -       -
                                                    6
                                                    7
                                                    8

      AMWC    9       INP        15     0/ 8        5       -       -       -
                                                    6
                                                    7
                                                    8

     S2MIO   10       INP        16     0/ 8        4       -       -       -
                                                    5
```

292009-9

					6 7 8			
A12	11	INP	–	–	5 6 7 8	–	–	–
A13	14	INP	–	–	5 6 7 8	–	–	–
A19	23	INP	–	–	4 5 6 7 8	–	–	–

OUTPUTS

Name	Pin	Resource	MCell #	PTerms	Feeds: MCells	OE	Clear	Clock
RAM0CS	15	CONF	8	2/ 8	–	–	–	–
RAM4KCS	16	CONF	7	2/ 8	–	–	–	–
RAM8KCS	17	CONF	6	2/ 8	–	–	–	–
RAM16KCS	18	CONF	5	2/ 8	–	–	–	–
EPROMCS	19	CONF	4	1/ 8	–	–	–	–
80C88CLK	20	COIF	3	1/ 8	–	–	–	2
82C84RDY	21	RONFA	2	1/ 8	–	–	–	–
STOPCLK	22	RORFA	1	1/ 8	3	–	–	–

UNUSED RESOURCES

Name	Pin	Resource	MCell	PTerms
–	1	–	–	–
–	3	–	9	8
–	4	–	10	8
–	13	–	–	–

PART UTILIZATION

81% Pins
87% MacroCells
9% Pterms

292009-10

June 1986

3

Implementing a CMOS Bus Arbiter/Controller in the 5C060 EPLD

DANIEL E. SMITH
APPLICATIONS ENGINEERING
INTEL CORPORATION

Order Number: 292012-001

IMPLEMENTING A CMOS BUS ARBITER/ CONTROLLER IN THE 5C060 EPLD

CONTENTS

INTRODUCTION

This application note shows how to implement a CMOS Bus Arbiter/Controller in an Intel 5C060 EPLD (Erasable Programmable Logic Device). The note includes a brief overview of a similar circuit implemented with typical PLA devices, a more detailed discussion of the 5C060 implementation, and a summary.

The bus priority resolution and arbitration scheme selected for the circuit is that used by the industry-standard MULTIBUS I interface. Operation and timing for the MULTIBUS I interface is well understood by most engineers and is described in readily available Intel publications. Thus, a description of the MULTIBUS I interface is not included here. The bus arbiter/controller functions shown here support both serial and parallel priority resolution between bus masters. Timing is equivalent to MULTIBUS I specifications. Electrical specifications for both the PLA and EPLD approaches vary from MULTIBUS I standards. Neither of the two circuits discussed here provide the full current sink capability for **all** MULTIBUS I signals. Because the EPLD implementation is designed for CMOS systems, however, this requirement is not relevant for the 5C060 implementation.

PLA APPROACH

The functional equivalent of a MULTIBUS I arbiter/controller can be implemented in two 20-pin PLA-type devices as shown in Figures 1 and 2. (Figure 1 shows the logic for the arbiter device. Figure 2 shows the logic for the controller and the connections to the arbiter.) Figure 3 shows the arbiter list file as an example of PLA-type files. Two different 20-pin PLA devices are required to implement the arbiter and controller functions, a 16R4-type device and a 16L8-type device.

Implementation of logic devices in PLA-type devices, such as those shown here, has proven to be quite beneficial. Development time and cost is much less than for custom silicon device designs. The two PLA-type devices take up less board space than a discrete TTL implementation of the same functions. In addition, the two raw devices can also be used for different functions in other products, thereby reducing inventory costs. As a result of these factors (and others), use of PLA-type devices has grown substantially in recent years.

With the increased density and flexibility of EPLD devices over typical PLA-type devices, even greater space, inventory, and cost savings can be obtained by using EPLDs. The following section shows an implementation of the same arbiter/controller functions in a single 24-pin 5C060 EPLD device.

5C060 IMPLEMENTATION

The equivalent functions for both the MULTIBUS I arbiter and controller fit inside a **single** 5C060 EPLD device. The 5C060 device is available in a 24-pin 0.3" DIP package. Figures 4 and 5 show logic diagrams for the arbiter and controller functions. When compared with the PLA implementation, some differences in the design are immediately apparent. These differences result from the characteristics of the EPLD macrocell or from corrections to the circuit used in Figures 1 and 2.

The major change resulting from the EPLD macrocell structure concerns the EPLD output buffers. Since output buffers from macrocells are non-inverting (PLA-type devices typically contain inverting buffers), signals enter the buffers in the same logic orientation from which they are to appear at the output. The logic for the EPLD (shown in Figures 4 and 5) incorporates this change.

Some errors in the PLA-type implementation have also been corrected in the EPLD design. These changes are as follows:

- The M/$\overline{IO}$ input to the MRDC/ and MWTC/ gates is inverted. M/$\overline{IO}$ distinguishes between memory and I/O cycles. The PLA-type implementation does not use this signal properly; the PLA-type controller generates read or write commands to both memory and I/O at the same time, which can result in contention between memory and I/O during bus transfers.
- BPRO/ is gated by BPRN/ in the EPLD design. When using serial priority resolution, this allows the highest priority arbiter to prevent all other masters from controlling the bus. (In the PLA design, BPRO/ is enabled/disabled only by a local request. Higher priority arbiters cannot disable all other arbiters. This can result in contention between bus masters. By gating BPRO/ with BPRN/ in the EPLD design, this source of bus contention is prevented.)

Figure 6 shows the list file for the arbiter/controller device. Figure 7 shows the report file produced by the iPLDS software. This file contains a pinout diagram of the final programmed device and provides a resource usage map for the device.

Most of the input and output signals are self-explanatory to those familiar with Intel processors and the MULTIBUS I interface. The XREQ input is the bus transfer request signal from the address decode logic. The BUSY/ and CBRQ/ outputs are bi-directional, simulated open-collector outputs. These outputs use the iPLDS 5C060 (Combinational-Output I/O-Feedback) primitive in the list file. The BUSY/ signal serves to illustrate this use of EPLD outputs.

A pull-up resistor is used externally (i.e., on the backplane) to hold BUSY/ high when no arbiter is in control of the bus. When the arbiter is granted control of the bus, AEN is clocked high, which enables the output of the BUSY/ driver. Since the input to the BUSY/ driver is low during normal operation (RESET/ inverted), the enabled driver pulls BUSY/ low to signal other arbiters that the bus is in use. When the arbiter is finished using the bus, AEN goes low to disable the BUSY/ driver (three-state output). The pull-up resistor pulls BUSY/ high to signal other arbiters that the bus is free for use if needed.

Note that BUSY/ is also routed into the bus grant logic as input BSI. BSI prevents the arbiter from taking control of the bus (and driving BUSY/ low) when some other arbiter already has control of the bus. Thus only one arbiter may pull BUSY/ low at any one time.

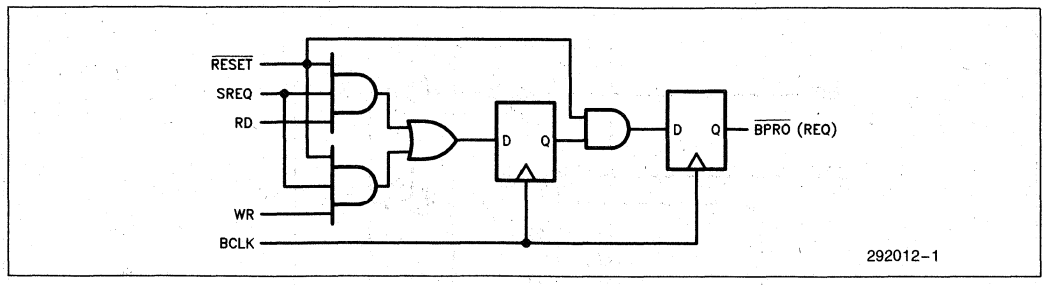
The one difference between standard MULTIBUS I logic levels and the EPLD implementation described here relates to the BCLK/ signal. MULTIBUS I bus arbitration uses the negative-going edge of BCLK/ to synchronize events. All 5C060 flip-flops, however, clock on the positive-going edge of BCLK/. If all bus masters in the system use the same arbiter implementation, this poses no problem. Otherwise, an external inverter is required for the BCLK/ input.

COMPARISON/SUMMARY

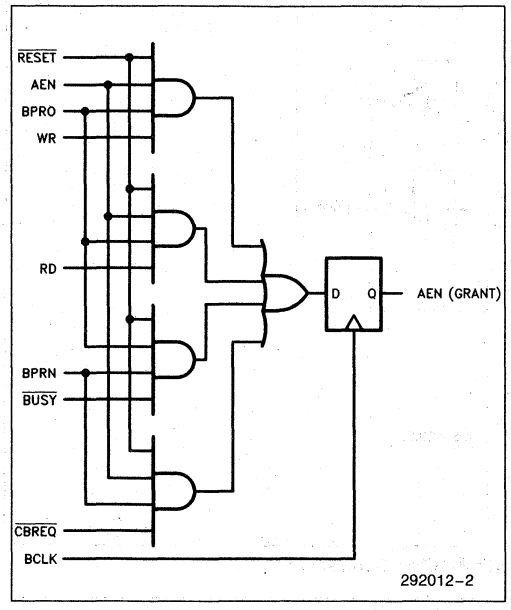
Both the PLA and EPLD implementations of the bus arbiter/controller result in a lower device count than a discrete logic circuit. Lower device count means less p.c. board space, fewer assembly steps, and fewer device interconnects. Both PLA and EPLD implementations are quicker and less expensive to develop than a custom gate array or dedicated silicon device.

In contrast to the PLA approach, however, the EPLD implementation requires only a single device, while the PLA approach requires two different devices. Thus the EPLD approach results in twice the cost savings (inventory and assembly) and half the programming activity to produce the device. Fewer device interconnects also means greater reliability. In addition, programmed EPLD devices can be erased and reprogrammed for a different application if needed, a feature not available with PLAs.

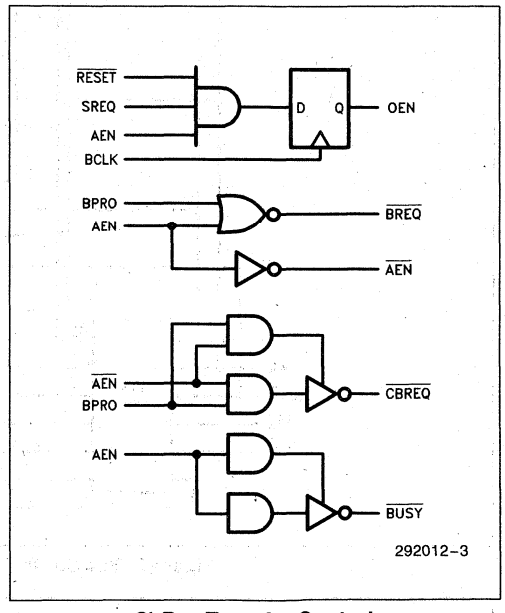
Overall, the greater flexibility, and the incremental design, manufacturing, and cost advantages of EPLD devices make them ideal for many applications where PLA devices would otherwise be used.



A) Request Synchronizer



B) Grant/Access Logic



C) Bus Transfer Control

Figure 1. PLA Approach to a Bus Arbiter

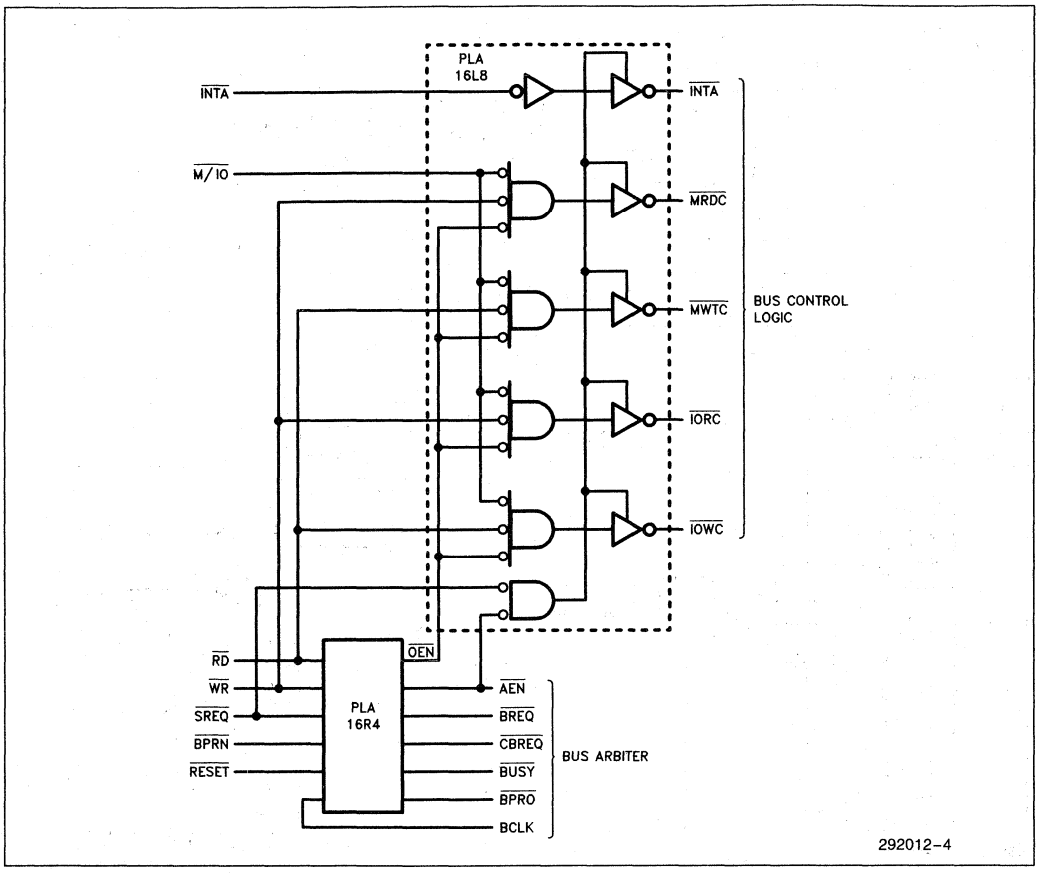
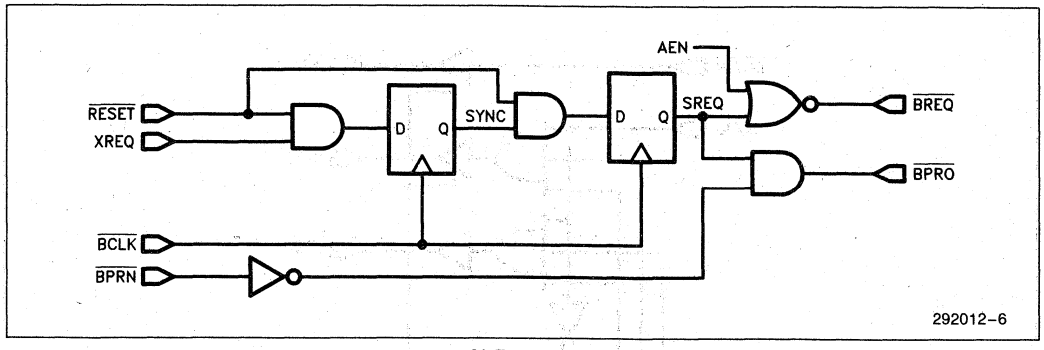


Figure 2. Bus Controller with Arbiter Connected

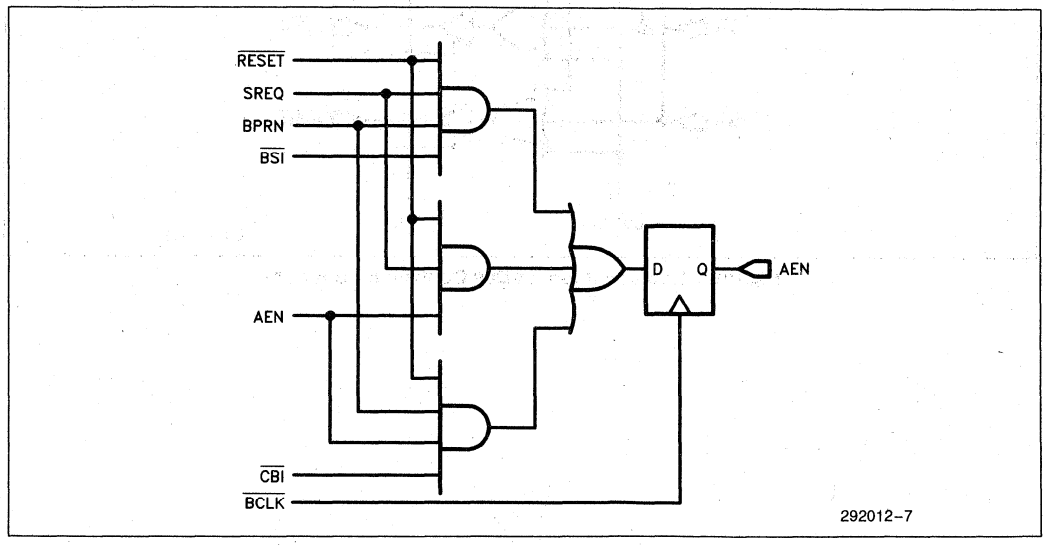
```
PLA16R4                                          PLA DESIGN FILE
ARB001                                        D. E. ENGR.  1/1/85
MULTIBUS I ARBITER
SOME SYSTEM COMPANY
BCLK  /WR    /RD   /SREQ /RESET /BPRN  NC    NC    NC  GND
/E    /CBREQ /BUSY /SYNC /BPRO  /AEN   /OEN /BREQ NC  VCC

SYNC  := /RESET*SREQ*WR          +
         /RESET*SREQ*RD

BPRO  := /RESET*SYNC

AEN   := /RESET* AEN*BPRO*WR     +
         /RESET* AEN*BPRO*RD     +
         /RESET*BPRO*BPRN*/BUSY  +
         /RESET* AEN*BPRN*/CBREQ

OEN   := /RESET*SREQ*AEN

IF(BPRO*/AEN) CBREQ = BPRO*/AEN

IF(AEN) BUSY = AEN

BREQ   = BPRO +
         AEN
```

292012-5

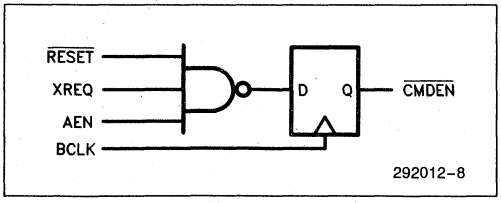
Figure 3. List File for PLA Arbiter



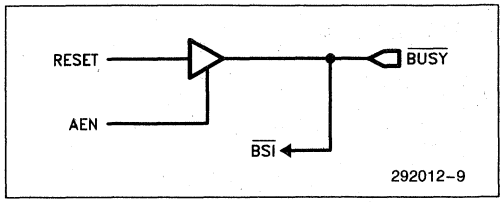
A) Request



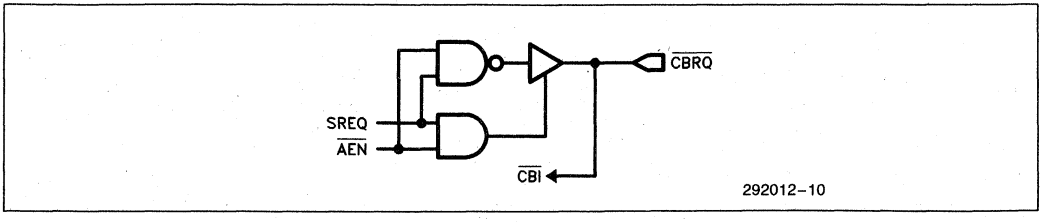
B) Grant



C) Command Enable



D) Busy



E) CBRQ

Figure 4. Logic Diagram of Bus Arbiter Functions

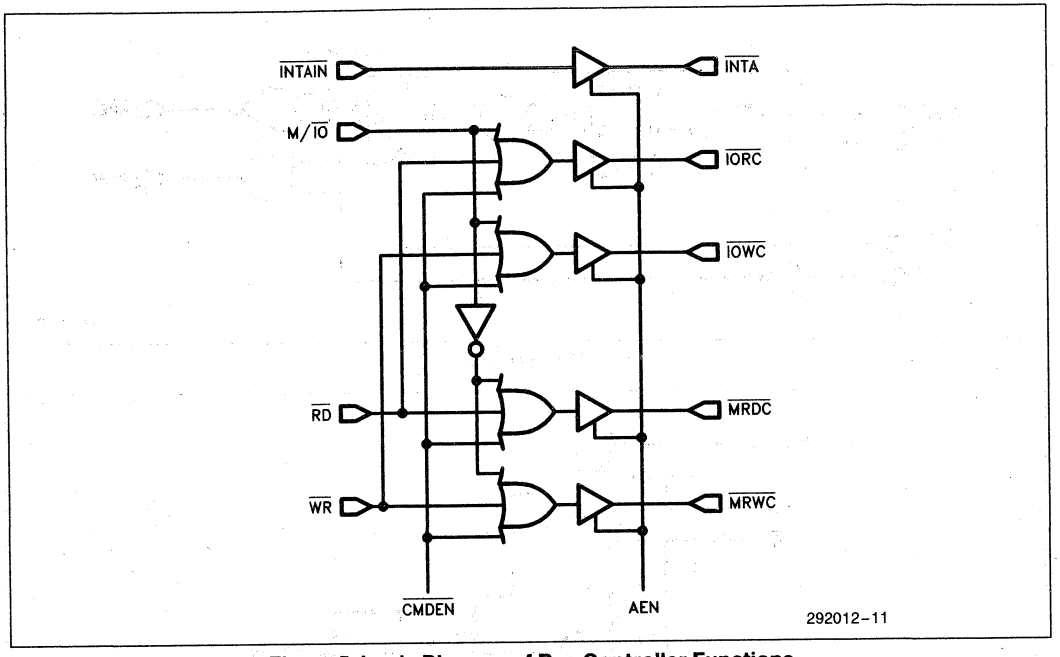


Figure 5. Logic Diagram of Bus Controller Functions

```
DANIEL E. SMITH
INTEL CORPORATION
MARCH 27, 1986
VERSION 1.1
REV. A
5C060
CMOS BUS ARBITER/CONTROLLER


PART:     5C060
INPUTS:   BCLK, XREQ, RESET, BPRN, MIO,  RD,   WR,   INTAIN
OUTPUTS:  BPRO, AEN,  BREQ,  CBRQ, BUSY, INTA, MRDC, MWTC, IORC, IOWC


NETWORK:

BCLK     = INP (BCLK)                    %BUS CLOCK INPUT%
INTAIN   = INP (INTAIN)                  %INT. ACK. INPUT%
XREQ     = INP (XREQ)                    %SYSTEM REQUEST INPUT%
RESET    = INP (RESET)                   %RESET INPUT%
BPRN     = INP (BPRN)                    %BUS PRIORITY INPUT%
MIO      = INP (MIO)                     %MEMORY/IO INPUT%
RD       = INP (RD)                      %READ INPUT%
WR       = INP (WR)                      %WRITE INPUT%
BPRO     = CONF (BPROc,VCC)              %BUS PRIORITY OUTPUT%
AEN,AEN  = RORF (AENd,BCLK,GND,GND,VCC)  %ADDRESS ENABLE (GRANT)%
BREQ     = CONF (BREQc,VCC)              %BUS REQUEST%
CBRQ,CBI = COIF (CBRQc1,CBRQc2)          %CBRQ/ -- SIMULATED O.C.%
BUSY,BSI = COIF (BUSYc,AEN)              %BUSY/ -- SIMULATED O.C.%
INTA     = CONF (INTAIN,AEN)             %INT. ACK. OUTPUT%
MRDC     = CONF (MRDCc,AEN)              %MEMORY READ COMMAND%
MWTC     = CONF (MWTCc,AEN)              %MEMORY WRITE COMMAND%
IORC     = CONF (IORCc,AEN)              %I/O READ COMMAND%
IOWC     = CONF (IOWCc,AEN)              %I/O WRITE COMMAND%
SREQ     = NORF (SREQd,BCLK,GND,GND)     %VALID BUS REQUEST%
SYNC     = NORF (SYNCd,BCLK,GND,GND)     %SYNCHRONIZED REQUEST%
CMDEN    = NORF (CMDENd,BCLK,GND,GND)    %COMMAND ENABLE%
```

292012-12

```
EQUATIONS:

BPROc    = (SREQ * /BPRN);
AENd     = RESET * SREQ * /BPRN * BSI +
           RESET * SREQ * AEN +
           RESET * /BPRN * AEN * CBI;
BREQc    = /(SREQ + AEN);
BUSYc    = /RESET;
CBRQc1   = /(SREQ * /AEN);
CBRQc2   = SREQ * /AEN;
MRDCc    = /MIO + RD + CMDEN;
MWTCc    = /MIO + WR + CMDEN;
IORCc    = MIO + RD + CMDEN;
IOWCc    = MIO + WR + CMDEN;
SREQd    = RESET * SYNC;
SYNCd    = RESET * XREQ;
CMDENd   = /(RESET * XREQ * AEN);


END$
```

292012-13

Figure 6. iPLDS Network List File

```
Logic Optimizing Compiler Utilization Report

 ***** Design implemented successfully

DANIEL E. SMITH
INTEL CORPORATION
MARCH 27, 1986
VERSION 1.1
REV. A
5C060
CMOS BUS ARBITER/CONTROLLER


                5C060
              _ _ _ _ _
        BCLK -| 1    24|- Vcc
         MIO -| 2    23|- XREQ
   RESERVED -| 3    22|- INTA
   RESERVED -| 4    21|- IOWC
   RESERVED -| 5    20|- IORC
         AEN -| 6    19|- MWTC
        BPRO -| 7    18|- MRDC
      INTAIN -| 8    17|- BUSY
          WR -| 9    16|- CBRQ
          RD -|10    15|- BREQ
        BPRN -|11    14|- RESET
         GND -|12    13|- GND
              - - - - -


**INPUTS**

                                                                  |          Feeds:
      Name   Pin   Resource   MCell #    PTerms                  |  MCells     OE   Clear   Clock

      BCLK     1        INP         -         -                  |        -      -       -    CLK1

       MIO     2        INP         -         -                  |        2      -       -       -
                                                                  |        3
                                                                  |        4
                                                                  |        5

    INTAIN     8        INP        14      0/ 8                  |        1      -       -       -

        WR     9        INP        15      0/ 8                  |        2      -       -       -
                                                                  |        4

        RD    10        INP        16      0/ 8                  |        3      -       -       -
                                                                  |        5

      BPRN    11        INP         -         -                  |       12      -       -       -
                                                                  |       13

     RESET    14        INP         -         -                  |        6      -       -       -
                                                                  |        9
                                                                  |       10
                                                                  |       11
                                                                  |       12

      XREQ    23        INP         -         -                  |        9      -       -       -
                                                                  |       10
```

292012-14

Figure 7. iPLDS Report File

```
**OUTPUTS**

      Name  Pin  Resource   MCell #    PTerms  :  MCells   Feeds: OE  Clear  Clock
       AEN    6      RORF        12       3/ 8  :      7          -7      -      -
                                                       8           1
                                                       9           2
                                                      12           3
                                                                   4
                                                                   5
                                                                   6
      BPRO    7      CONF        13       1/ 8  :      -          -      -      -
      BREQ   15      CONF         8       1/ 8  :      -          -      -      -
      CBRQ   16      COIF         7       1/ 8  :     12          -      -      -
      BUSY   17      COIF         6       1/ 8  :     12          -      -      -
      MRDC   18      CONF         5       1/ 8  :      -          -      -      -
      MWTC   19      CONF         4       1/ 8  :      -          -      -      -
      IORC   20      CONF         3       1/ 8  :      -          -      -      -
      IOWC   21      CONF         2       1/ 8  :      -          -      -      -
      INTA   22      CONF         1       1/ 8  :      -          -      -      -


**BURIED REGISTERS**

      Name  Pin  Resource   MCell #    PTerms  :  MCells   Feeds: OE  Clear  Clock
                      3      NORF         9       1/ 8  :      2          -      -      -
                                                       3
                                                       4
                                                       5
                      4      NORF        10       1/ 8  :     11          -      -      -
                      5      NORF        11       1/ 8  :      7          7      -      -
                                                       8
                                                      12
                                                      13


**UNUSED RESOURCES**

      Name  Pin  Resource     MCell     PTerms
         -   13         -         -          -

**PART UTILIZATION**

95%     Pins
100%    MacroCells
11%     Pterms
```

292012-15

Figure 7. iPLDS Report File (Continued)

intel®

APPLICATION NOTE

AP-321

November 1988

Fitting the 5C180

TODD KOELLING
PROGRAMMABLE LOGIC APPLICATIONS
INTEL CORPORATION

Order Number: 292053-001

FITTING THE 5C180

CONTENTS

INTRODUCTION

In many ways, fitting the 5C180 is like climbing a mountain. Just when what appears to be the summit is reached, another summit is revealed behind it. This may occur several times before the actual summit is surmounted.

Likewise, fitting a 5C180 may have several false summits. Just when one has conquered what appears to be the "problem", another problem often appears behind it. This may occur several times before the design fitting is complete.

This application note addresses the problems that can be encountered when trying to fit a 5C180 and offers suggestions on how to get past them. The key to the climb is examining what resources are still available after the software* complains that a particular resource is not available.

SUMMIT NUMBER ONE: PIN ESTIMATE

Before keying in the design, it is best to estimate the I/O pin requirements. This is done by counting the total number of inputs to the device and outputs from the device.

PROBLEM: Not enough Input Pins

HELP: Run all synchronous clocks through Clock Buffers (CLKBs). Shared clocks may use the same CLKB output which may result in reduction from 4 CLK input pins to 1 CLK input pin (see Figures 1a &

*iPLS II ver. 1.1 or later is ESSENTIAL for 5C180 designs as the fitting algorithm was significantly improved with this release.
*iPLS II ver. 1.5 or later is HIGHLY RECOMMENDED as the error messages and Utilization Report Files were significantly enhanced with this release.

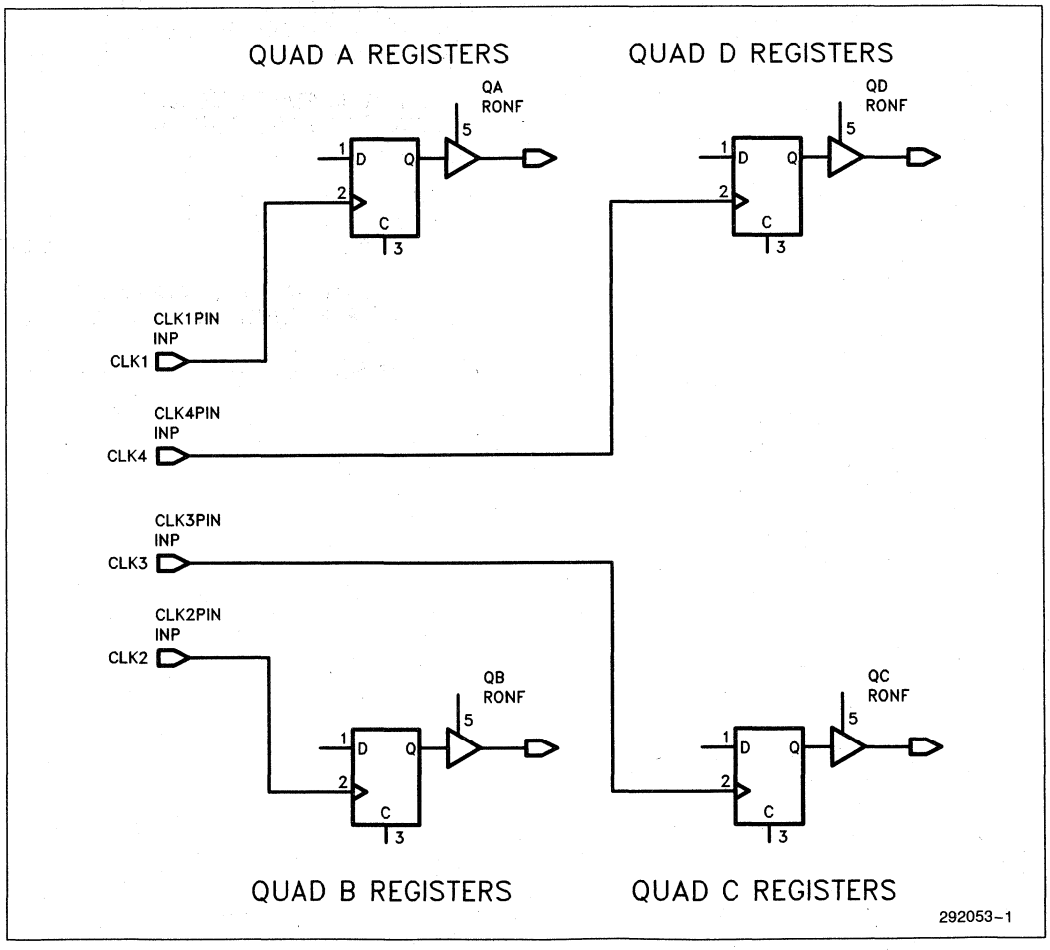


Figure 1a. Summit One—Input Clocks Before

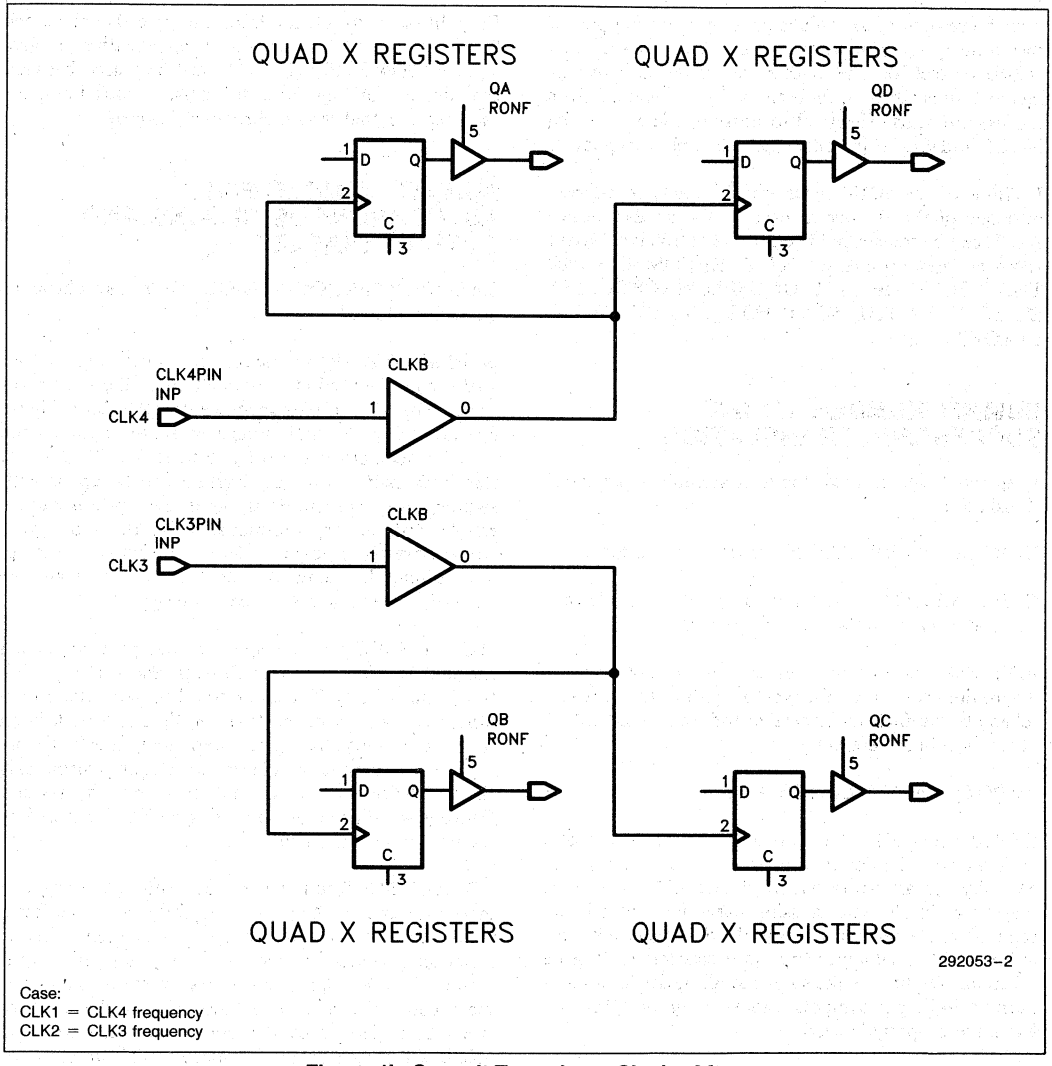


Figure 1b. Summit Two—Input Clocks After

1b). This also frees the registers that the clock feeds from the synchronous clock pin quadrant, increasing the chance of fitting later on. THIS PRACTICE IS RECOMMENDED FOR ALL DESIGNS.

PENALTY: Input setup time is shortened. (See Synchronous vs. Asynchronous A.C. Characteristics in Data Sheet).

If clock buffering cannot solve the problem, the design must be repartitioned to reduce the number of input pins. Repartitioning is explained in the next section.

SUMMIT NUMBER TWO: MACROCELL ESTIMATE

If the I/O pin requirements can be met, the next step is to consider the macrocell requirements. The total macrocell count can be estimated by counting the number of outputs plus the number of internal registers.

PROBLEM: Not enough macrocells

REPARTITIONING: Unless the fundamentals of the design can be changed, this error means that the design

must be repartitioned. This is done by removing part of the circuitry and placing it in a second device such as a 5C060 or 5C090. The 5C060 and 5C090 are recommended since their architectures (and therefore their ADFs) are nearly identical to those of the 5C180 (the NOCF and COCF primitives are the only exceptions).

Portions of the 5C180 ADF can be easily transferred into one of the smaller devices or the smaller device ADFs can be transferred back to the 5C180 if sufficient room is freed up later on. IT IS RECOMMENDED THAT FOUR OR FIVE UNUSED MACROCELLS BE LEFT IN THE 5C180 FOR USE BY LATER STAGES.

SUMMIT NUMBER THREE: SUCCESSFUL TRANSLATION

With the design entered, the next summit is successful translation.

ERROR: ***ERR-MAC-No macrofunction for: ...

EXPLANATION: The Macro Expander Module cannot find a macro for a network element.

FIX: Make sure correct search path is available for macro libraries. Check for typo or syntax error. If using schematic capture, make certain that only valid EPLD library symbols were entered.

ERROR: Any "***ERROR-XLT-..."

EXPLANATION: The Translator found a problem with the way the design was entered. These errors are basically syntax errors which violate ADF format. It may be a simple typo, missing parenthesis or missing semicolon. Remember that the iPLS II LOC does differentiate between upper and lower case letters. If using schematic capture, make sure that all device inputs and outputs have pin symbols and that all the pins and wires are properly labeled.

FIX: Refer to your iPLS II manual or call the EPLD Hotline, 1-800-323-EPLD, for help on the tough ones.

SUMMIT NUMBER FOUR: REGISTER CLOCK INPUTS

ERROR: ***ERROR-XLT-Clock input must be driven by INP or CLKB

EXPLANATION: The clock for a flip-flop must be driven synchronously by a direct quadrant clock pin input (INP) or asynchronously through a Clock Buffer (CLKB). This problem occurs when an equation or gate logic is connected directly to the register clock input.

FIX: In order to tell the LOC software that the clock for a flip-flop will be driven by an equation or gate logic, a Clock Buffer (CLKB) must be placed between the equation or logic and the register clock input for each register that is asynchronously clocked.

SUMMIT NUMBER FIVE: ASYNCHRONOUS CLOCKS AND OUTPUT ENABLES

ERROR: ***ERROR-XLT-OE with asynchronous clock not allowed

EXPLANATION: Asynchronous clock and output enable can't be used at the same time in the same macrocell. The 5C180 basic macrocell architecture, Figure 2, shows why. A single p-term is shared between the asynchronous clock and the output enable. This means that both switches in the diagram can be up or both switches can be down. By trying to use a p-term output enable with an asynchronous clock, the top switch would have to be down while the bottom switch is up. This cannot be done as then the register would be clocked and enabled with the same signal.

WORKAROUND: To get around this problem, one of the signals must be routed through another macrocell (see Figures 3a-b). The clock could be generated in another macrocell, sent out to a pin, then sent back in on the synchronous clock pin. Alternately, in a first macrocell the register is placed as an asynchronously clocked NORF. In a second macrocell, the register feedback is sent out to a pin using a CONF enabled by the desired enable signal.

PENALTIES: Routing the clock through a separate macrocell and back in offers slightly better performance—since the synchronous clock to ouput time is faster than a second macrocell delay, but this implementation uses a lot of resources—three pins and two macrocells. The second method, routing the feedback from the register back and controlling the output enable in a second macrocell is more straightforward and uses less resources.

SUMMIT NUMBER SIX: GREATER THAN ONE PRODUCT-TERM REGISTER CONTROLS

ERRORS: ***INFO-FIT- Eqn. too big, 4/-1 PTerm(s), on OE signal OE3

***INFO-FIT- Illegal inversion of CLEAR input (CLR1)

EXPLANATION: As shown in the basic 5C180 macrocell architecture, Figure 2, only one product term (multiple input AND gate) is available for the register

clock, clear, and output enable. This means that any control resource containing an OR gate following Boolean minimization will not fit. Likewise, any control resource requiring an invert will not fit either. To find the offending signal, LOOK AT THE EQUATIONS SECTION OF THE LOGIC EQUATION FILE (.LEF).

WORKAROUND: Once the offending signal has been located, it must be routed through another macrocell using an NOCF primitive (see Figure 4a-b). If the control signal is a clock, then a clock buffer (CLKB) must also be added.

PENALTY: Unless a trick explained below can be used, this routing results in the use of an additional macrocell and a doubling of the signal propagation delay.

Clr Fitting Trick

PROBLEM: Register clear input breaks 1 p-term resource limit

TRICK: If register has D input of either VCC or GND, substitute SR Flip-Flop.

EXPLANATION: D-type EPLD register has only 1 AND gate feeding CLR; SR Flip-Flop utilizes logic array for CLR input allowing a max of 8 AND gates (p-terms) for the CLR resource.

PENALTIES: SR Flip-Flop is synchronously clocked. D register has asynchronous clear.

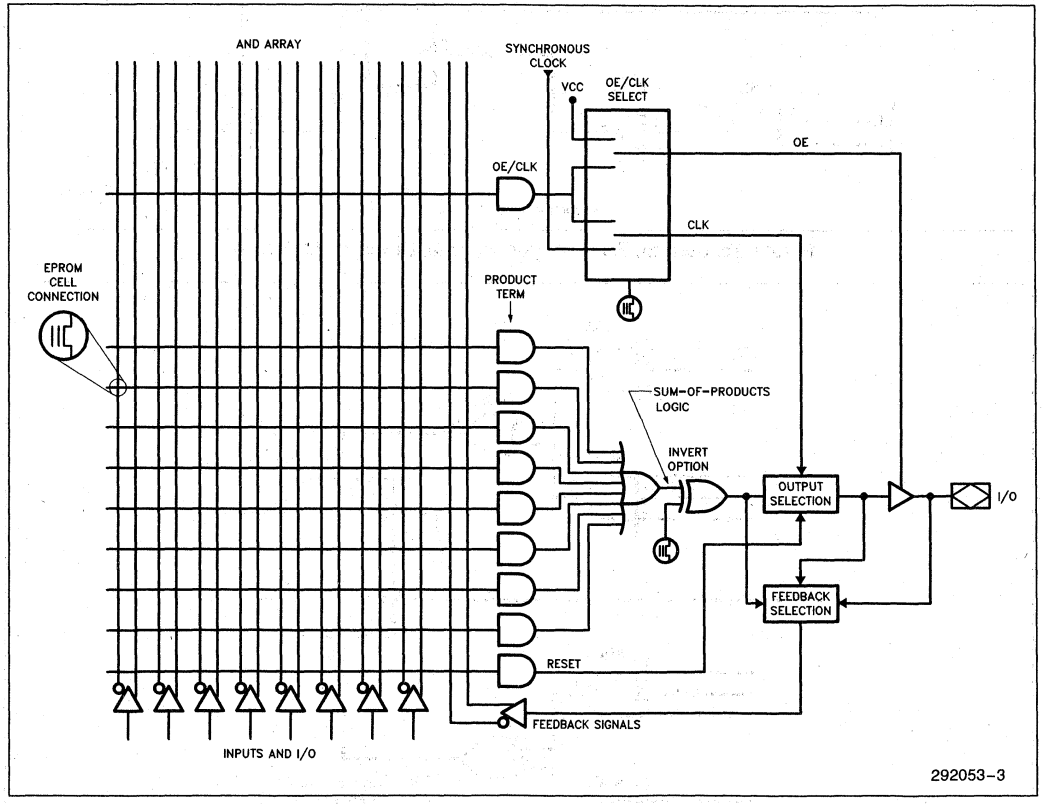


Figure 2. Basic Macrocell Architecture of the 5C180

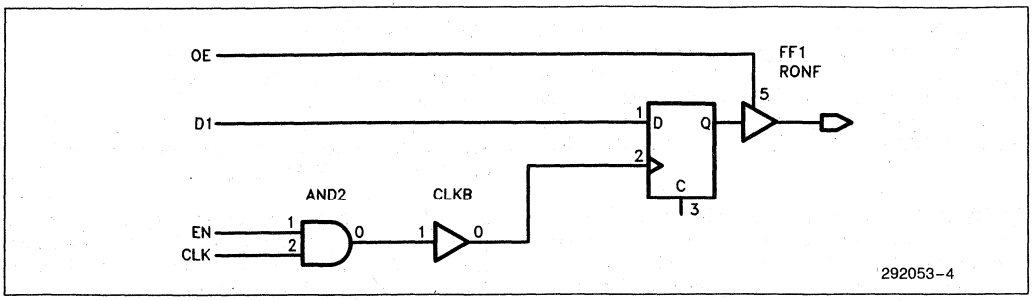


Figure 3a. Summit Five—Asynchronous Clock and OE Before

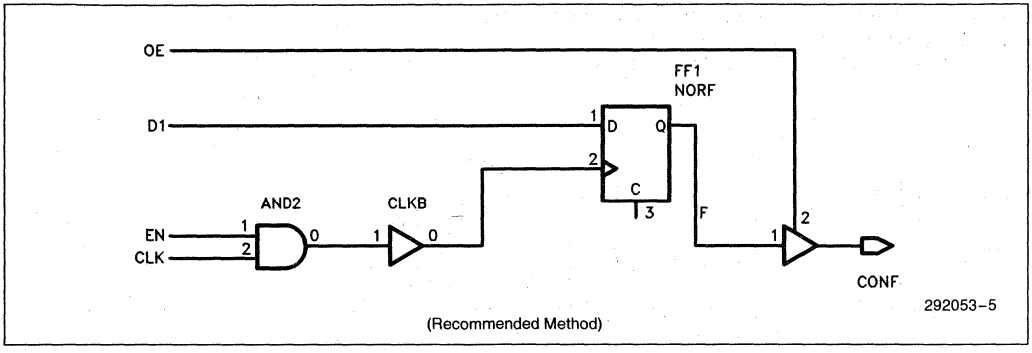


Figure 3b. Summit Five—Asynchronous Clock and OE After

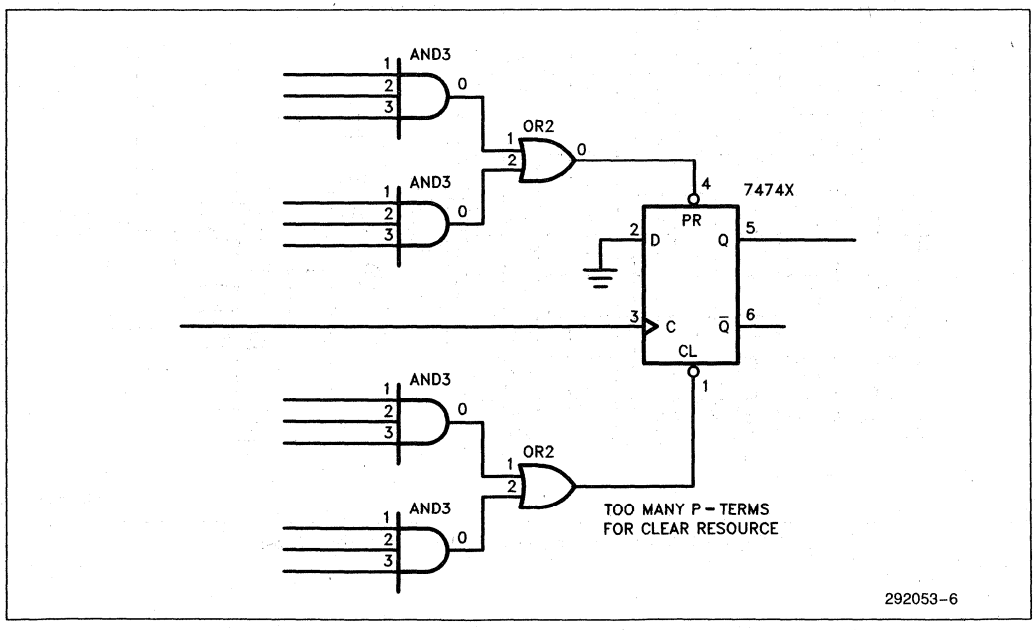


Figure 4a. Summit Six—Too Many P-Terms on Control - Clear

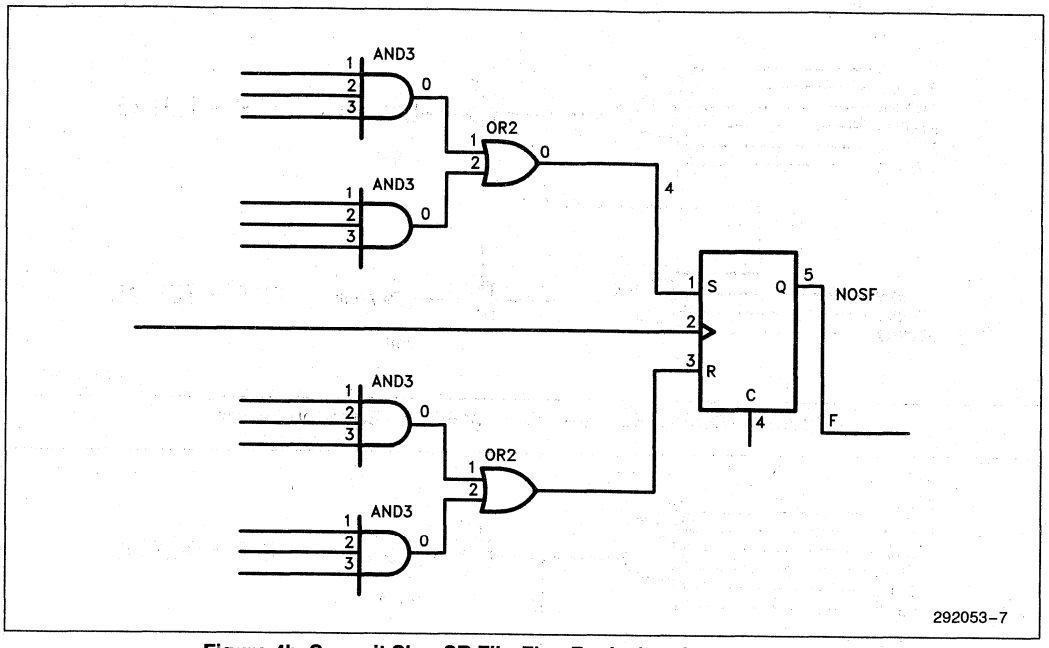


Figure 4b. Summit Six—SR Flip-Flop Equivalent Implementation

OE Fitting Trick

PROBLEM: Output enable on an equation or combinational equation exceeds 1 p-term resource limit.

TRICK: If a low output rather than a tri-state output can be tolerated, the signal can be gated rather than tri-stated.

EXPLANATION: Run the OE and the equation through an AND gate before going to a pin. The output of the pin will only follow the equation when the enable is active, otherwise it will be zero.

PENALTY: Forced low rather than tri-state output.

SUMMIT NUMBER SEVEN: NOT ENOUGH P-TERMS FOR AN EQUATION

ERROR: ***INFO-FIT- Too many PTerms to fit in any MCell: 10/8 for EQN.

EXPLANATION: Since the 5C180 has a maximum of eight product terms per macrocell, there's a chance that this number may be exceeded by the requirements of an equation. If so, the equation is cited by the LOC and can be examined by looking at the EQUATIONS section of the .LEF.

WORKAROUND: The workaround for this situation may already be in place! If any portion of the logic (or equation) is routed into an NOCF or CONF elsewhere in the design, that feedback can be taken and routed into the equation (see Figure 5a-b). This means a single feedback node—rather than several nodes will now feed the equation and thereby reduce the p-term count. (If the feedback is to be taken from a CONF primitive, the CONF must be changed to a COCF or COIF to make the feedback available.)

If part of the logic or equation is not routed into a NOCF or CONF elsewhere in the design, then part of the equation must be routed through a NOCF, COCF, or COIF primitive. A NOCF is recommended as it does not use a pin if placed in a global macrocell. If several equations are in violation of the eight p-term maximum, try to choose a group of logic that is common to all of the equations. In this manner, the p-term count for several equations can be brought down with the use of single extra macrocell, rather than the use of a macrocell for each equation.

PENALTY: Any time a portion of an output signal must be routed through another macrocell a speed penalty is incurred (roughly one propagation delay). If an already existing macrocell can be found, then there is no architectural penalty. If a new one must be created, then another macrocell is added to the total macrocell count.

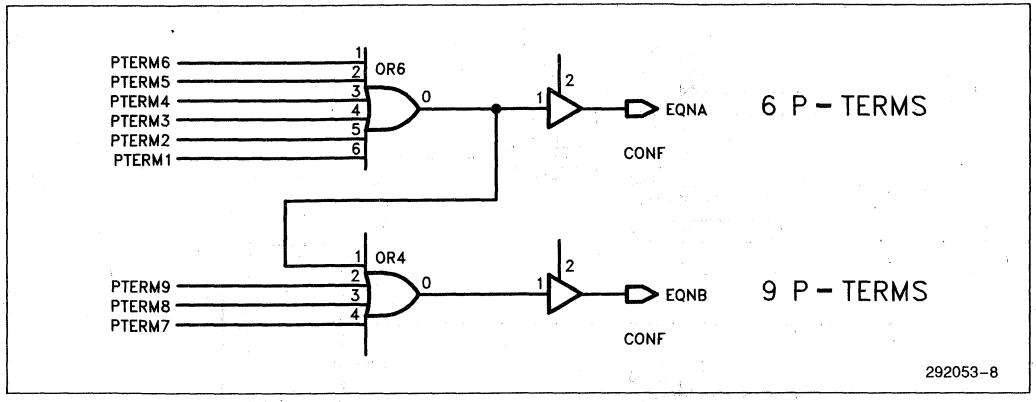


Figure 5a. Summit Seven—Too Many P-Term Equation Before

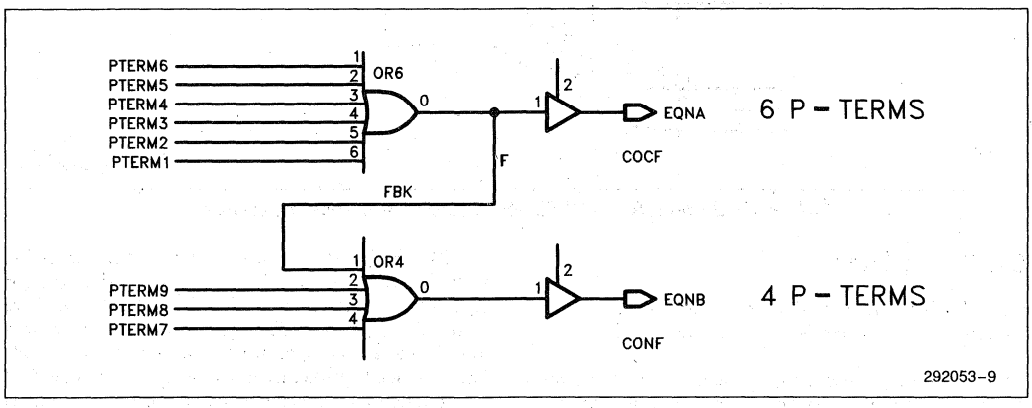


Figure 5b. Summit Seven—Too Many P-Term Equation After

SUMMIT EIGHT: MACROCELL RESOURCES EXCEEDED

ERROR: ***INFO-FIT- Design requires too many macrocells

EXPLANATION: If this error didn't occur at the beginning, there's a good chance summits five, six or seven will push the macrocell count over the limit. (Remember that the macrocell count includes not only the outputs, but also the buried resources such as NOCFs, NORFs and NOTFs). To find out exactly how many macrocells the design requires, LOOK AT THE NETWORK: SECTION OF THE LOGIC EQUATION FILE (.LEF). The inputs list in the LEF will list both the outputs and all the buried resources required by the design. If the count exceeds 48, then too many macrocells are required.

FIX: Repartition. The same applies if the number of input pins is exceeded.

THE FINAL ASCENT: NOT ENOUGH GLOBAL FEEDBACK!

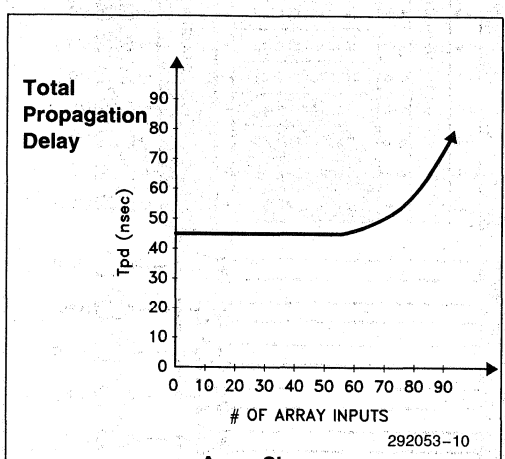
Congratulations! If you have made it this far, you have demonstrated courage, intelligence and tenacity beyond that of the average climber. You will soon be rewarded, but first there is one more obstacle to be overcome. Welcome to the North Face of local/global feedback!

A Word About Local/Global Feedback

First of all, why does local/global feedback exist? The answer can be found in the graph shown in Figure 6. The propagation delay versus array size is shown for the 5C060/090/180 family. As the number of inputs into the array increases, the propagation delay increases...exponentially. If all the inputs and feedback were made global, the 5C180 would have 136 inputs feeding each array (remember that both true and complement polarities must be fed into the array of a PLD architec-

ture). This would have put the 5C180 Tpd in the 250 - 300 ns range! By making eight macrocells local for four quadrants, the number of array inputs was dropped to 88 and the Tpd subsequently decreased to 75 ns.

The tradeoff to the local/global routing scheme is more difficult design routing. With the help of the iPLS II and a couple of tricks, however, most designs can still be fit.



Array Size

Device	Array Inputs	T_{PD}
5C060	40	45 ns
5C090	72	50 ns
5C180	88	75 ns

Array size increases Capacitance
Capacitance increases Propagation Delays

Figure 6. Propagation Delay vs. Array Size for the 5C060/090/180 Family

A Few Notes

The global/local macrocell assignments are shown in Figure 7. Please note that:

1. Dedicated input pins are GLOBAL.
2. Global macrocell I/O pin are GLOBAL.
3. Global macrocell internal feedback paths are LOCAL.
4. Local macrocell pin/feedback paths are LOCAL.

where GLOBAL means that the signal feeds all macrocells and LOCAL means that the signal only feeds the macrocells in its quadrant.

Clock Input Pins

The clock input pins feed the global bus like the regular inputs, except the synchronous register input connection is dedicated to a particular quadrant. Thus, each clock input can be used as a logic input in all quadrants or a clock input in its own quadrant. To be used as a register clock input in a quadrant outside its own, however, it must be tapped from the global bus via an asynchronous clock buffer (CLKB).

Global Macrocell Feedback

The feedback path is local for GLOBAL macrocells while the I/O input is global for all GLOBAL macrocells. Thus, changing the feedback of a register or combinational equation from a standard feedback to I/O pin feedback path will change the routing from local to global. The iPLS II LOC automatically recognizes and performs this through a process called "promoting". With the promotion process, global routing can be obtained on signals that would otherwise remain local.

***INFO-FIT- Promoted "TEQNF" from NOCF to COIF

Burying a Register in a Global Cell

Because the global macrocells have separate register and I/O pin feedback paths, it is possible to "bury" a register or equation by disabling the output buffer and still use the pin as an input. The iPLS II LOC automatically assigns an input to the pin of a buried register macrocell if it is necessary and possible. Such assignments are documented in the Utilization Report File (.RPT). If manual assignment is desired, it may be performed by placing the input pin assignment in the ADF INPUTS: list and assigning the buried register feedback to the same pin in the OUTPUTS: list (Figure 8). Registers or equations can only be buried on global macrocells, since local macrocells only have one feedback path that is used for either the register or the pin feedback.

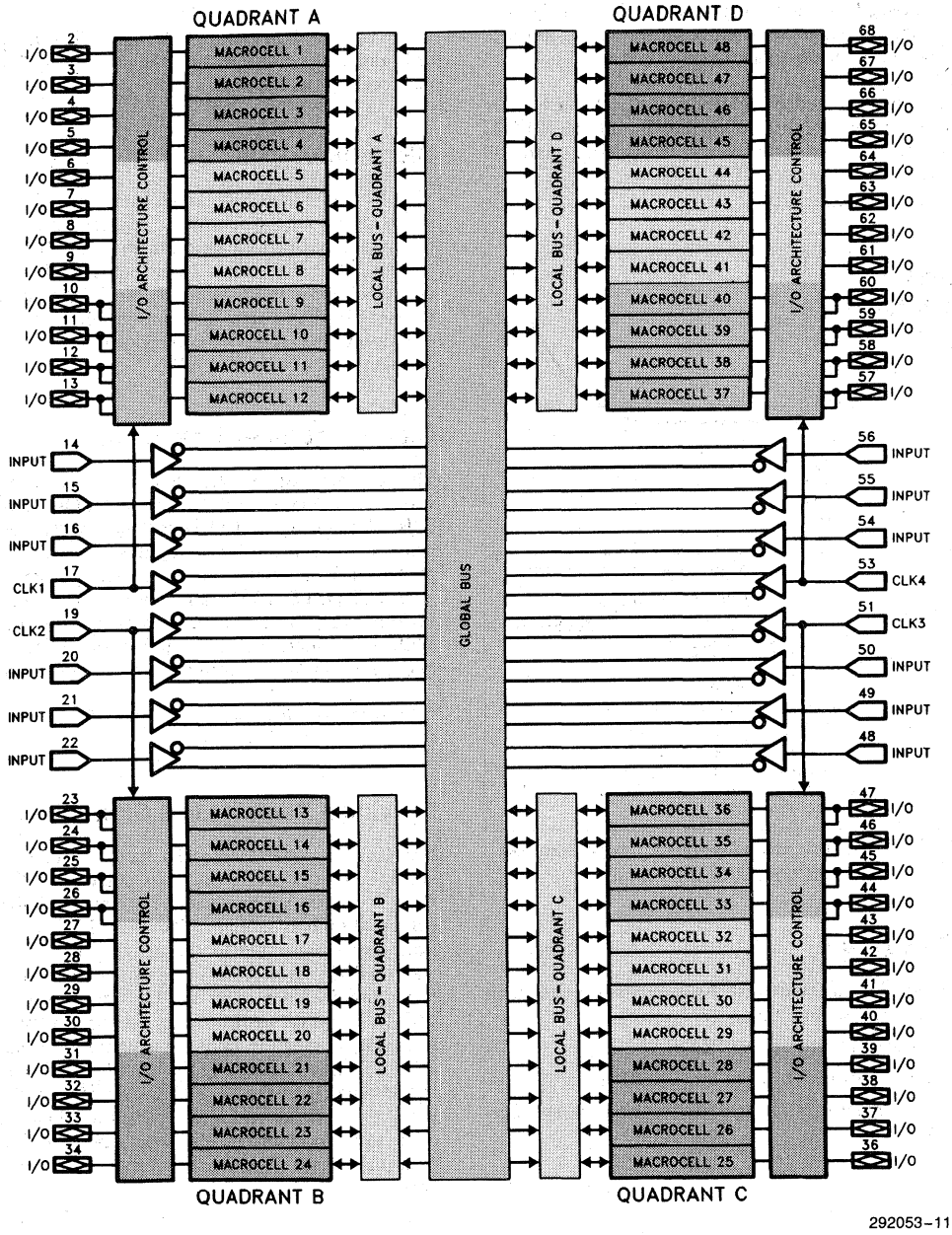
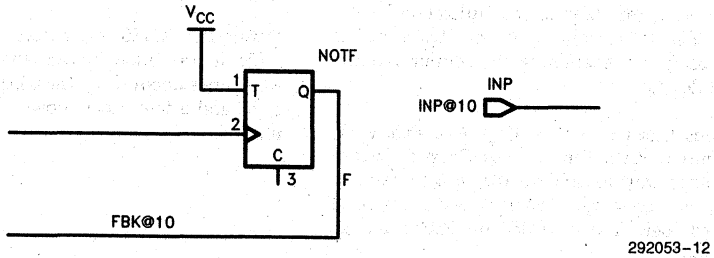


Figure 7. 5C180 Block Diagram



292053-12

Buried Register Pin Assignment in ADF

```
Intel
PLDO Apps
July 27, 1988
5C180 Buried Reg Pin Assignments

PART: 5C180
INPUTS: A@15, B@10, CLK@17  % Assign input B to pin 10 %
OUTPUTS: FBK@10             % Assign buried reg feedback FBK %
                            % to pin 10 (GLOBAL macrocell 9) %
NETWORK:

A = INP(A)                  % Inputs %
B = INP(B)
CLK = INP(CLK)

FBK = NORF(IN,CLK,GND,GND)  % Buried Register %

EQUATIONS:

IN = A * B * FBK;           % Register Input Equation %

END$
```

Figure 8. Assigning Buried Reg in Schematic

Two Global Fitting Tricks

If the LOC is unable to fit the design, there are a couple of manual tricks that may help:

PROBLEM: NOT ENOUGH GLOBAL FEEDBACK

RESOURCES AVAILABLE: EXTRA MACROCELLS

TRICK: Duplicate the macrocell logic that needs to be global in two (or more) regions with appropriate renaming (see Figure 9).

EXPLANATION: This makes the signal available in two regions via two local macrocells rather than one which can't be global.

PENALTIES: There may be a slight timing discrepency between the two macrocells for combinational logic, but any discrepency will be small (< 2 ns).

PROBLEM: NOT ENOUGH GLOBAL FEEDBACK

RESOURCES AVAILABLE: EXTRA INPUT PINS

TRICK: Send out the signal that needs to be global and externally connect it to one of the input pins.

EXPLANATION: Inputs feed the global bus, making the signal available in all quadrants.

PENALTIES: An output buffer plus input buffer minus feedback delay is added (approximately 25 ns). An external connection must be made on the board.

NOTE:
For the previous tricks, look at the Utilization Report (.RPT) file. The "Interconnect Cross Reference" is particularly useful for examining the routing requirements of the design.

If the previous tricks cannot be done (see Figure 11) and scrutinization of the Interconnect Cross Reference reveals no other way to achieve the desired routing, repartitioning is necessary. That is, place a chunk of interconnected logic into a 5C060 or 5C090 and go back to the start.

CONCLUSION

Fitting the 5C180 is a process with many stages. One difficulty may hide the next and fixing one problem will sometimes uncover another. Equipped with the iPLS II LOC and a few tricks, however, fitting can be accomplished.

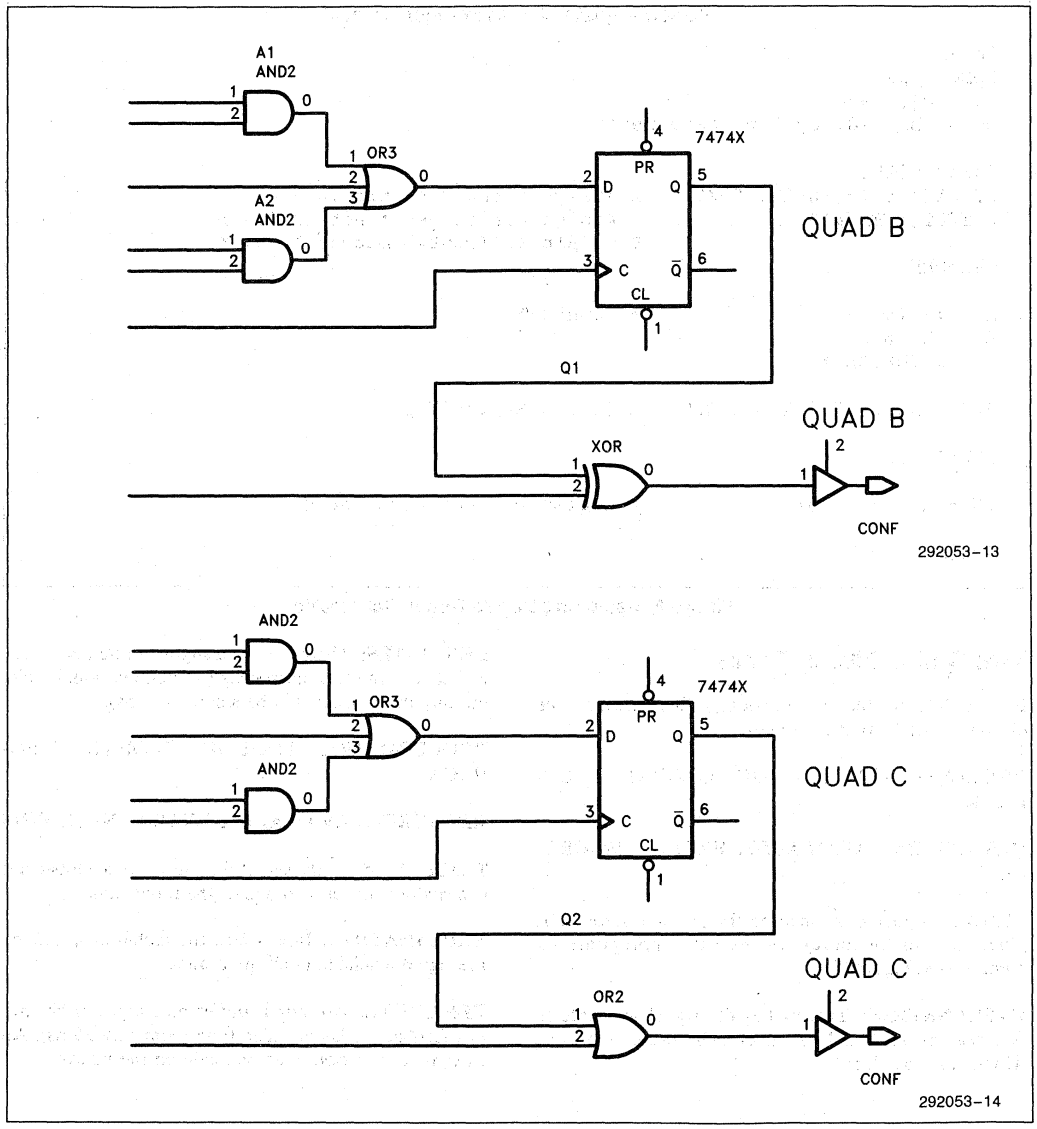


Figure 9. Not Enough Global Feedback Extra Macrocells—Fit

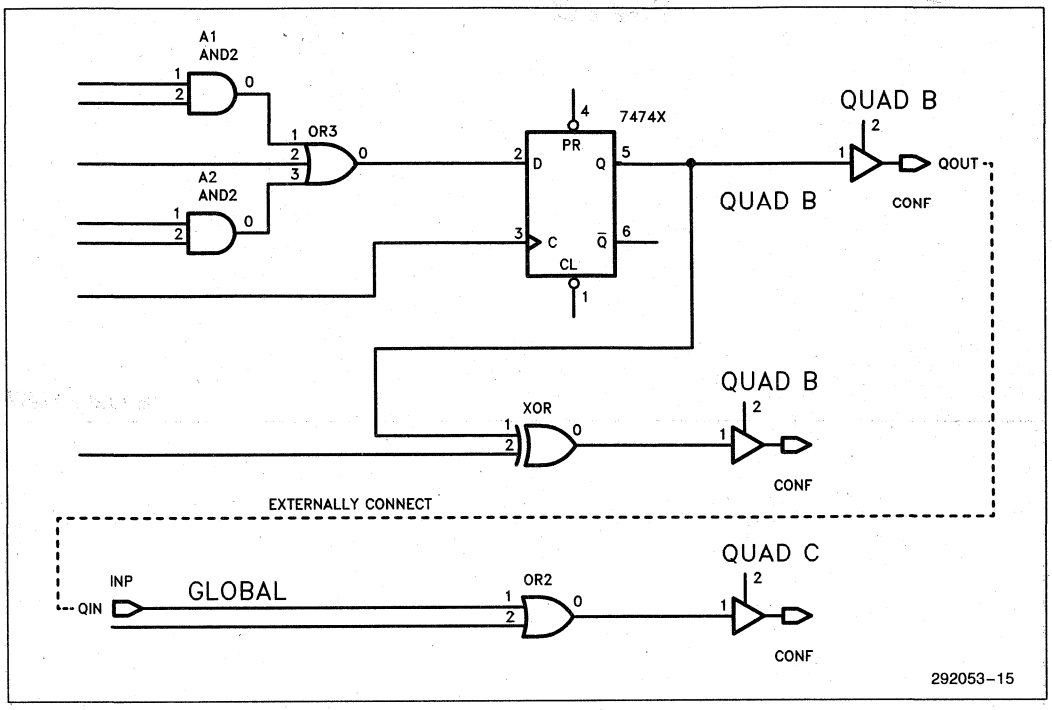


Figure 10. Not Enough Global Feedback Extra Inputs—Fit

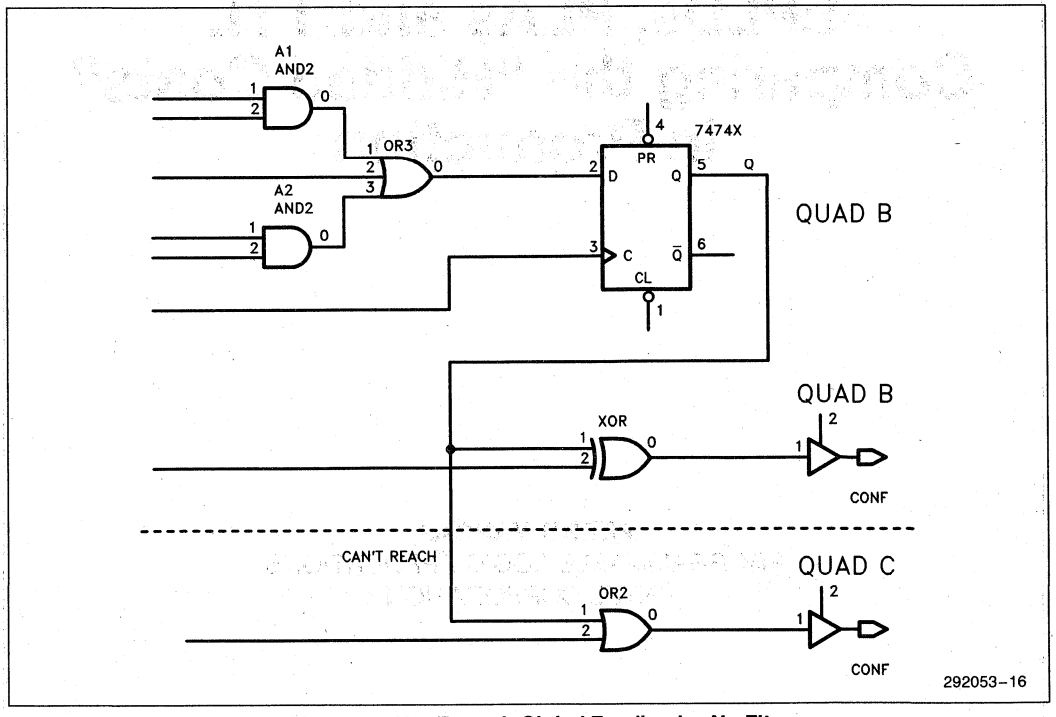


Figure 11. Not Enough Global Feedback—No Fit

APPLICATION NOTE

AP-307

January 1987

EPLDs, PLAs and TTL Comparing the "Hidden Costs" in Production

PEDRO VARGAS
PROGRAMMABLE LOGIC APPLICATIONS
INTEL CORPORATION

Order Number: 292030-001

EPLDs, PLAs AND TTL COMPARING THE "HIDDEN COSTS" IN PRODUCTION

CONTENTS

INTRODUCTION

When comparing logic alternatives, too often the outcome is dominated by the piece price of the components. A side by side comparison based on component costs only, may give the appearance that EPLDs are cost prohibitive. However, when the overall cost of manufacturing a system is considered, the higher integration of EPLDs proves to be a cost-effective solution.

OBJECTIVE

This application note examines the total costs associated with designing, prototyping, and manufacturing a system. Once these costs have been examined, a comparison is made between EPLDs and other logic alternatives. By being aware of these additional costs, the engineer can make a more accurate cost comparison as a design is begun.

COSTS DEFINED

Costs can be difficult to pinpoint, let alone measure. However, with a bit of examination, we can break down costs into the following categories;

- Design costs — the cost of conceiving a product
- Prototype costs — first implementation of the product idea
- Production costs — volume manufacturing of the product

Usually, the brunt of the cost for the first two categories is dismissed as NRE (non recurring expense). The effect of these costs on the overall project is examined later, let's look at the third category. Production costs, can be further broken down into;

- Component costs — the cost of the parts per board
- Inspection costs — labor costs for receiving the parts
- Inventory costs — the cost for storing, handling and dispensing the parts
- PCB fabrication — the cost for labor and equipment used in building a board
- Integration costs — the cost of harnesses, enclosures, nuts and bolts etc.

It's important to understand how the cost of a product is affected not only by the cost of the ICs used, but also by the other costs listed above. Figure 1 is a graph which shows this relationship.

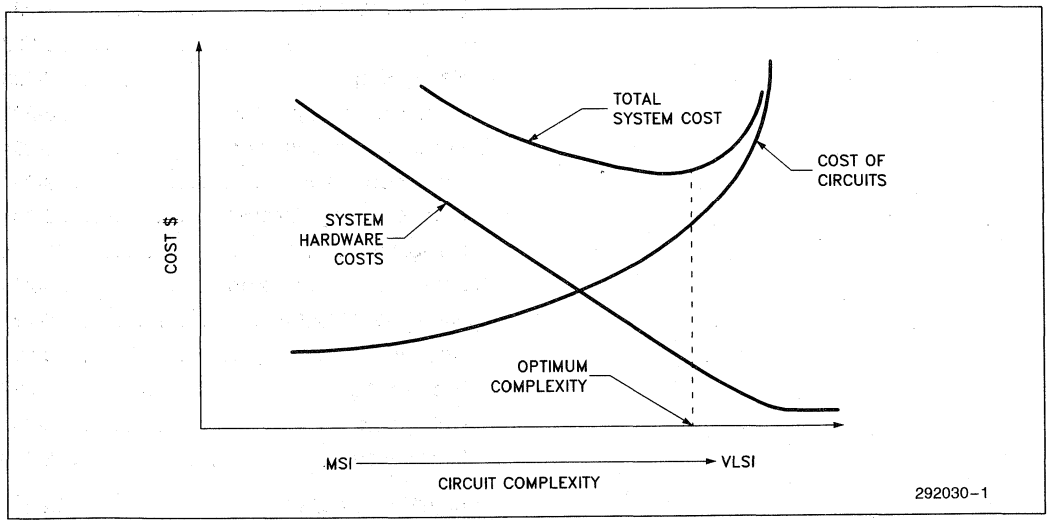


Figure 1. Optimizing Circuit Complexity

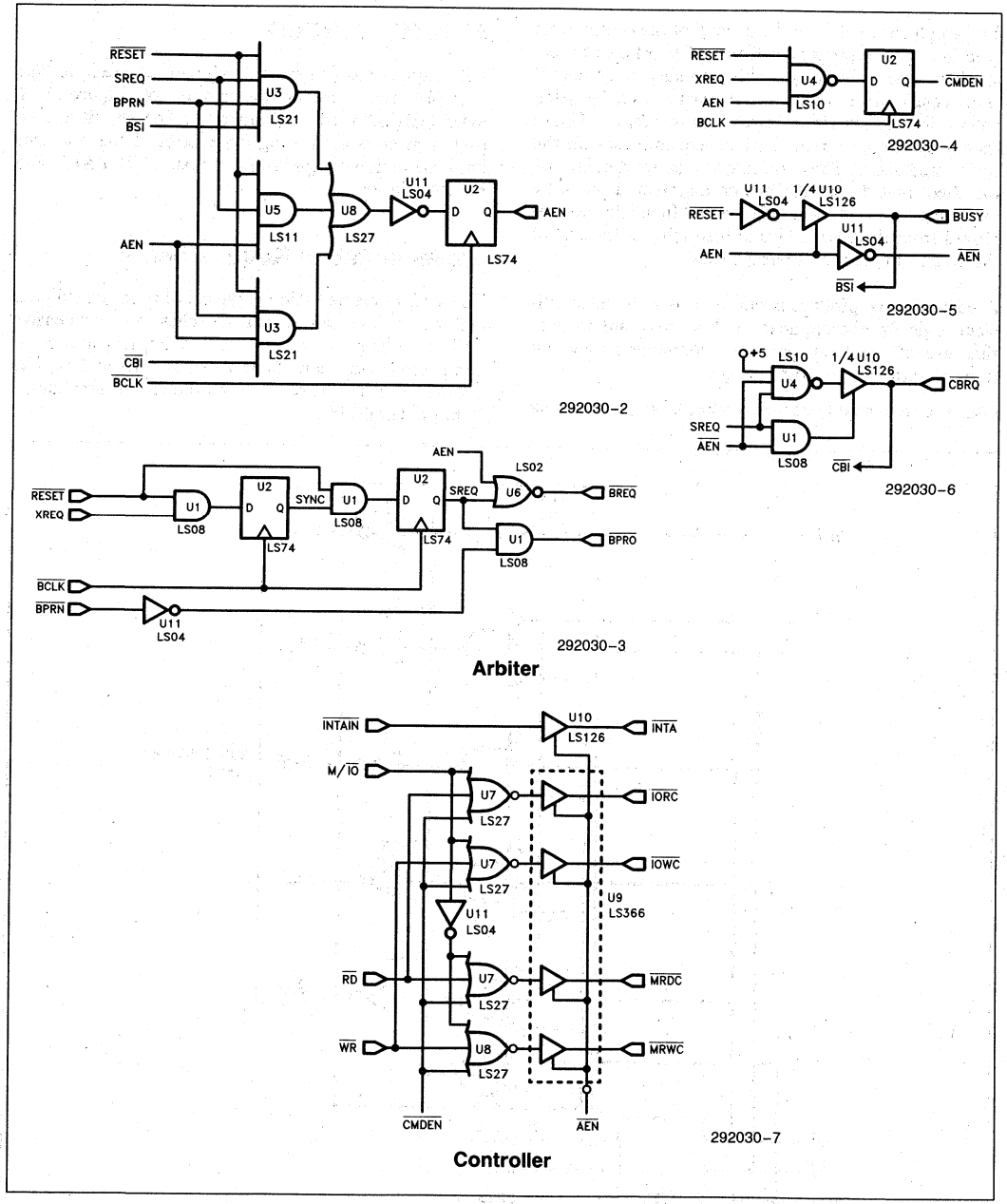


Figure 2. MULTIBUS Arbiter/Controller–TTL Implementation

The graph shows that as the density of the components used in a system progresses from SSI to VLSI, the cost for these devices increases. This isn't surprising, denser chips cost more to make. At the same time, by using denser devices, system hardware cost decreases. This is shown by the center line, which encompasses all the costs listed above. The bathtub curve above these shows the effect that denser ICs has on a system. That is, by using higher integration ICs, more functions are removed from the board. This in turn reduces the cost of the system in labor and parts costs.

A cost-effective product is one that uses the most efficient logic for the application. It's important to note that use of the least expensive component may not translate into system cost savings.

PAL* is a registered trademark of Monolithic Memories Inc.

ARBITER CIRCUIT

Let's explore costs in more detail with an example. The example used here is the circuit of Figure 2, a MULTIBUS® I arbiter/controller. The circuit is used by bus masters arbitrating for control of the bus. Our implementation comparison contrasts TTL, PAL*, and EPLD solutions.

Implementation Requirements

The TTL implementation is typical of many board level designs in the sense that it relies on inexpensive LSTTL. Figure 2 shows that the implementation is composed of standard logic gates and D-latches. The component list in Table 1 shows the circuit breakdown in more detail.[20]

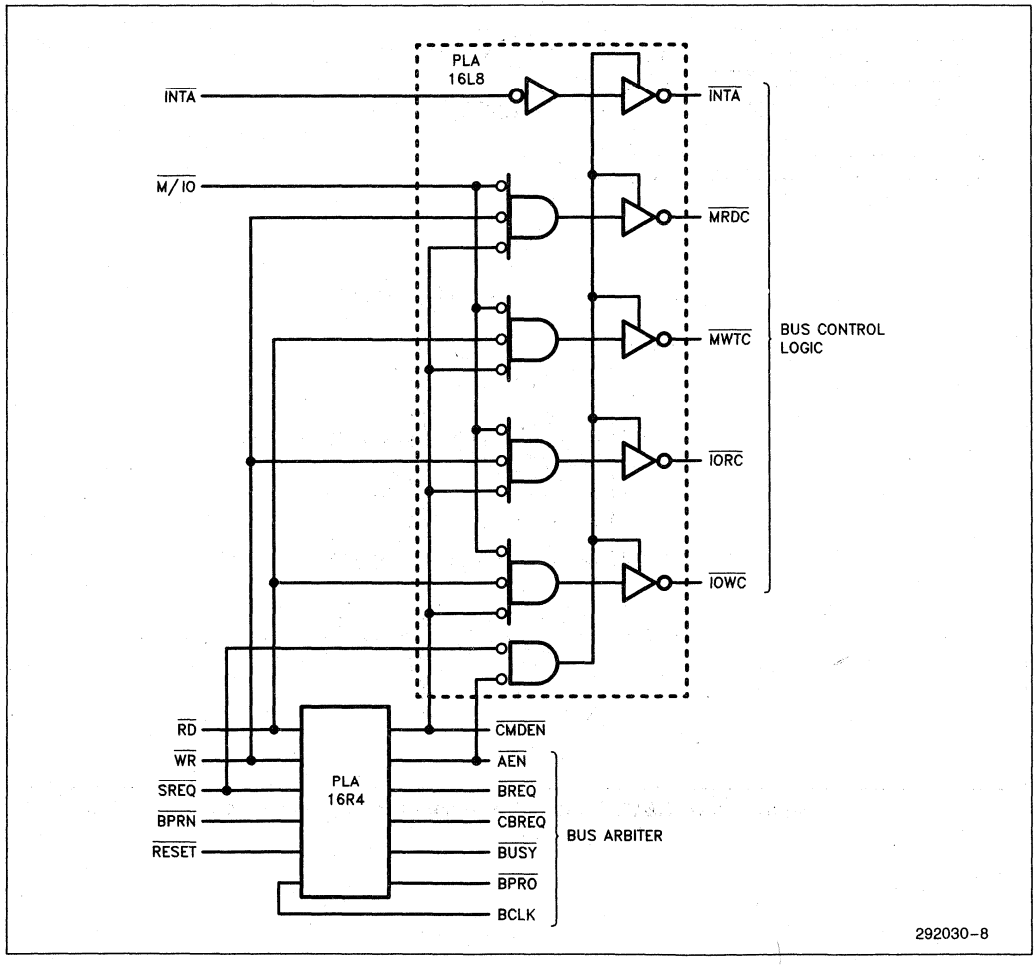


Figure 3. MULTIBUS Arbiter/Controller-PAL Implementation

Table 1. Arbiter/Controller TTL Component List

IC	Type	DIP	I_{CC} (mA)	Area (in^2)	Cost $
U1	LS08	14 PIN	8.8	0.21	0.18
U2	LS74	14 PIN	8	0.21	0.24
U3	LS21	14 PIN	4.4	0.21	0.22
U4	LS10	14 PIN	3.3	0.21	0.16
U5	LS11	14 PIN	6.6	0.21	0.22
U6	LS02	14 PIN	5.4	0.21	0.17
U7	LS27	14 PIN	6.8	0.21	0.23
U8	LS27	14 PIN	6.8	0.21	0.23
U9	LS366	16 PIN	21	0.24	0.39
U10	LS126	14 PIN	22	0.21	0.39
U11	LS04	14 PIN	6.6	0.21	0.16

The PAL version of the circuit is shown in Figure 3. Two PALs are used due to the requirement of registered outputs on several of the signals.[20]

The complete circuit can also be designed in one 5C060 EPLD (Figure 4).[18] Looking at the three figures quickly points out the amount of circuit board space required by each version. The three implementations are compared side by side in Table 2.

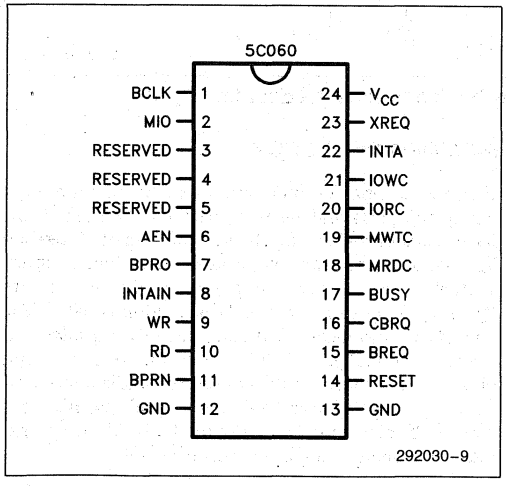


Figure 4. MULTIBUS Arbiter/Controller-EPLD Implementation

Table 2. Implementation Results for Arbiter/Controller

Item	TTL	PLA	EPLD
IC Count	11	2	1
Pin Count	156	40	24
Interconn	36	7	0
Area	2.34	0.6	0.36
I_{CC} (mA)	100	240	15
P_{wr} (mW)	500	1,200	75

- IC Count — The total chip count
- Pin Count — The total number of IC pins
- Interconnections — The traces required to connect logic gates together
- Area (inches-square)— The sum of the area of all ICs
- I_{CC} (mA) — The current consumed while active
- P_{wr} (mW) — Total power consumption at 5 VDC.

Production Costs

Earlier, we noted that production costs consist of many variables. Usually, these variables are lumped together under the term "hidden cost". Although hidden costs are kept in mind by engineers, lack of tangible figures usually precludes their use in detailed cost breakdowns. For this reason, several manufacturers and consulting firms have come up with typical costs per IC and per pin.

For example, SOURCE III (San Jose, CA) reports in one of their studies that the manufacturing cost of a system translates to about 0.35 cents per IC pin. ICE Corporation (Scottsdale, AZ) and EDN magazine concur that the inserted cost of an IC is about $2 dollars. DATAQUEST also published a cost of about $2 to $4 per IC. While the data seems to be consistent, most engineers want to see for themselves how figures like these might be arrived at. The next sections provide insight into this process.

COMPONENTS

The cost of the component is the easiest value to obtain. A quick call to a distributer or (at worst) a scan through the back of BYTE magazine (for TTL) gives us this cost. Table 3 shows the breakdown of component costs for each version of our MULTIBUS I circuit.

Table 3. Average Component Costs

Package	TTL	PLA	EPLD
DIP14	$0.25		
DIP16	$0.35		
DIP20	$0.55	$1.50	
DIP24		$2.90	$6.00

The price of TTL has changed very little for the last few years[24] while EPLDs are dropping in price tremendously. PALs have also leveled off in pricing. Why? Figure 5 shows the life cycle curve of IC products used by the semiconductor industry. From the curve we see that TTL is in the stable range and prices are not likely to drop much more. PALs are also maturing and approaching a stable pricing range. EPLDs however, are in a growth area and historically this is

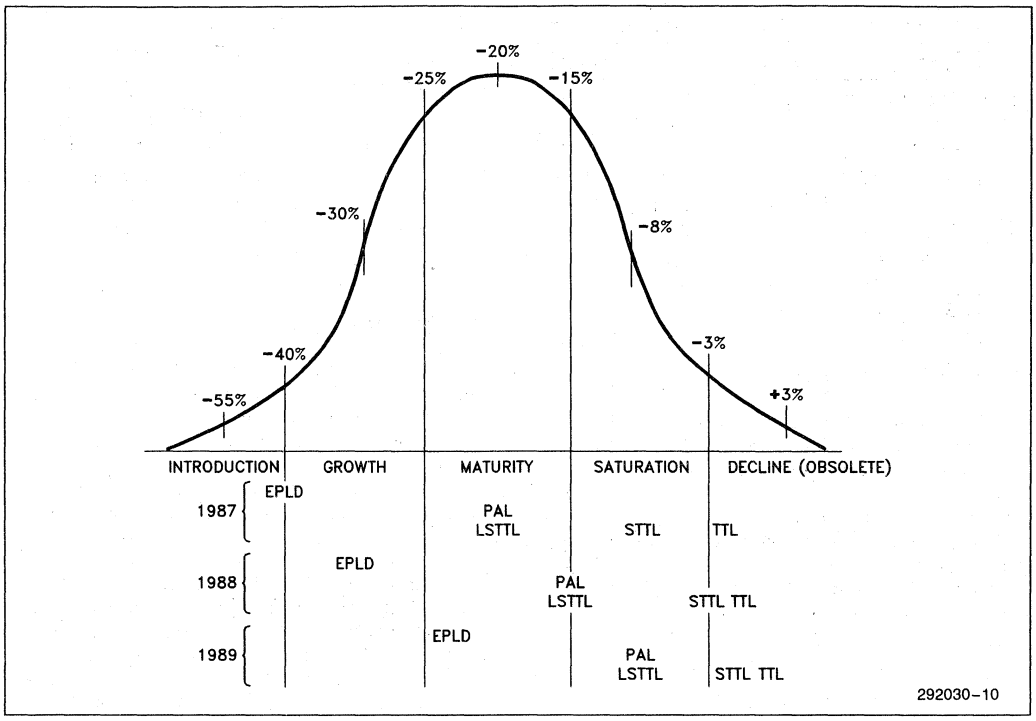


Figure 5. Typical Price Changes Through Semiconductor Product Life Cycle

where the heaviest pricing pressure is. This means that while EPLDs might be expensive (per part) right now, it's not out of the question to expect a 30% per year price reduction as the process is honed and perfected. In other words, it's also important to consider the price of a component at the projected production date, not just at design time.

Life cycle position is also important in understanding the gate cost that is associated with programmable logic devices like PALs and EPLDs. This relationship is shown in Figure 6. The curves translate our observation that newer devices have steeper price cuts during their introduction phase. The PAL curve shows that the cost per gate is leveling off due to the maturity of the device. In contrast, the EPLD is in the growth region, and based on the traditional price reductions, shows a cost per gate that intersects and bypasses the PAL curve.

INCOMING INSPECTION

For most companies, incoming inspection is more than taking the parts and putting them on the shelf. Most have visual checking as well as some form of IC testing. The variables here are, what amount of human intervention is needed, are automatic handlers needed, are "go/no go" tests or "binning" done automatically? The typical scenario means that components are graded and tested individually, and then placed into one of several bins or kitted. Because the operators handle a large variety of pinned devices (resistors, capacitors, ICs), the cost can be distributed on a per pin basis. Many companies use a penny per pin for this cost.[16]

Inspection cost = $0.01 per pin

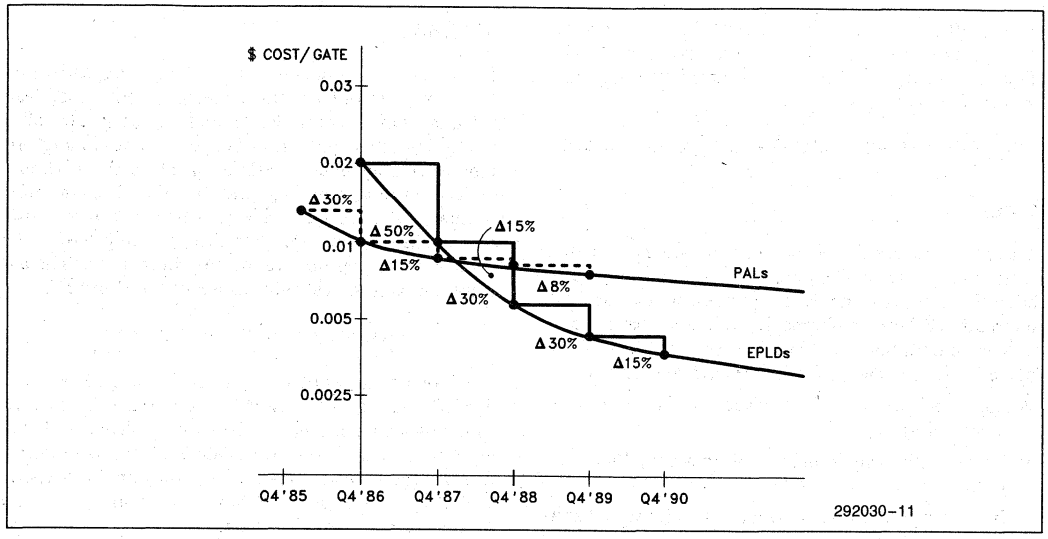


Figure 6. Projected Cost Per Gate

3

INVENTORY

While most engineers agree that reducing parts count on their board makes the cost of inventory less, they usually attribute this to the reduction in component costs alone. In reality, the overhead of carrying inventory is made up of the following factors;[21]

- Cost of the component
- Cost of storage
- Maintenance costs
- Data processing
- Usage
- Taxes insurance and interest
- Turnover rate

The American Production and Inventory Control Society (APICS) reports that since 1973 the median cost of carrying inventory has been about 25% of total production costs. They also note that the largest contributing factors are the cost of materials handling storage, and data processing. For simplicity, let's limit our inventory cost to these items.

Inventory cost = storage + maintenance + processing

Depending on the locale of a company, the cost of storage can vary greatly. However, this cost is charged on a square foot per year basis. Lets assume a conservative figure of $20 dollars and distribute this amoung the ICs in our example circuit.

storage = [Total IC area (sq. ft.) × $20]/IC count

Maintenance refers to the cost of handling, counting, marking, and auditing each IC. Each production manager has their own way of keeping tabs on this. One way is to charge on a per part basis. A review from several production oriented journals cites $0.3 cents as the typical handling charge for 16 pin devices.[23]

Maintenance = $0.03 per 16 pin part.

Processing[21] usually entails a parts log that tracks each part by manufacturer, cost, second source etc. Also, monthly shortage reports are quite common as are quarterly orders and audits. Limiting this cost to paper only, at one sheet of paper per week, per year, at a cost of a penny per part type;

Processing = $0.52 per part type per year

PCB FABRICATION

The cost of manufacturing (cutting, etching, drilling) a circuit board seems to vary around two pricing methods. Some fab houses charge on a square inch basis. Others base their price on a gut feeling based on previous jobs. The square inch method is the most common.

Items of interest in evaluating PCB costs are, number of ICs, number of traces and vias, and in general, the complexity of the board. Traces that are smaller than 10 mils require extra care in etching. Depending on complexity, and additional charge might be added to the area cost. This charge covers material loss in case of low etch yields. Yield is directly dependent on the number of ICs on a board. In other words, more ICs mean more holes, tighter traces, and a greater chance of losing some boards in their processing. The average going

rate is $0.20 cents per inch for double-sided boards. The price increases by about 40% for every two layers. This extra charge, however is too subjective to consider in our comparison.

PCB Fab = [$0.20 × total IC area (sq. inch)]/IC count

Traces

There is a real cost involved with traces, which doesn't surface until later in the production cycle or on a later board revision. A technical paper presented at the 1984 international Test Conference[1] estimates that the cost of a trace on a board is ten to thirty times that of one made in silicon. The cost of traces is taken up by:

- Increased drilling (more traces = more vias = more holes)
- Lower PCB yield (smaller mill lines drop the board yield)
- Increased risk of trace to trace shorts (lower reliability)
- More expensive artwork mods (it costs more to move traces around on a board)
- More expensive PCB mods (cost of cuts, jumpers, and rework)

In our circuit example, an extra trace is that which is unnecessary in contrasting implementations. For example, referring to Figure 2, of all the traces required to connect/RESET in the TTL implementation, only one will be required for the EPLD and PAL circuit (the input); the others won't be needed.

For our comparison, let's take the median value of twenty as our multiplying factor. Since a silicon trace costs an order of magnitude less than an EPLD gate ($0.01), the resulting cost of a PCB trace is;

($0.01/10) × 20 = $0.02 cents per trace

Trace cost = [total trace count × $0.02]/IC count

ASSEMBLY

The cost of assembling a board is largely dependent on labor charges and capital. Assembly consists of lead forming, component insertion, and soldering. The labor charge is hourly and varies between domestic and offshore assembly houses. While machines can certainly do lead cutting, crimping, and insertion, human intervention is still an expensive presence. Assembly costs can be charged on a per board or per chip basis. The latter is more appropriate for our comparison. The average charge (domestically) is about $0.10 per IC.

Assembly = $0.10 per 16 pin part

One important result of using high integration parts like EPLDs is that the assembly procedures (manual or automatic) go smoother. This is due to fewer parts being handled, and less overheating of the equipment. Overall, the industry reports less insertion faults (parts stuffed wrong) as denser ICs are used and as insertion equipment matures with them.

TEST

Test strategies can vary, but the typical test flow for a board[3] is shown in Figure 7. The process is basically taking a board through increasing complexity levels of testing. For example, ATE might be a bed of nails fixture that catches 60 percent of the faults. Test bed is usually a backplane with all boards known good except for the one under test. System test is the final integration of all the boards that were tested individually.

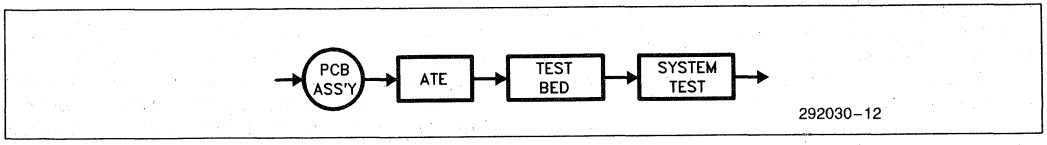


Figure 7. Typical Test Flow

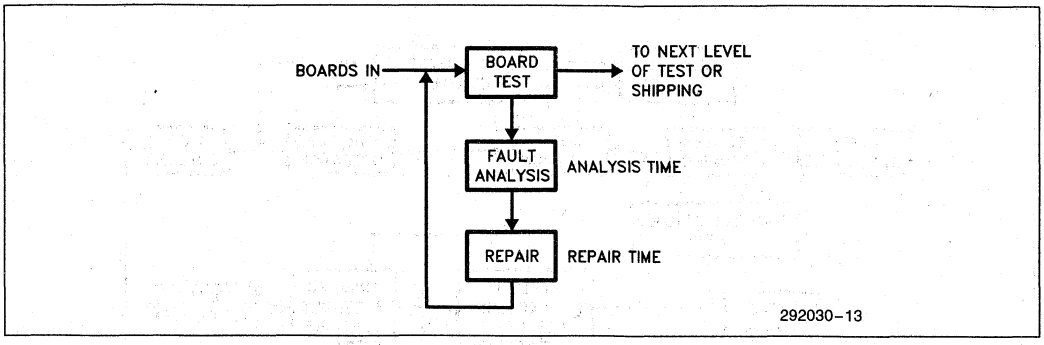


Figure 8. Typical Test and Repair Loop

Errors can occur at any step of the test flow; each time this happens, a test loop is initiated. This loop is depicted in Figure 8. The cost for testing a device depends on the cost of the equipment, depreciation, the labor rate, and other factors that are company dependent. There are several ways to reduce test costs, but the best way is to reduce the probability of errors occuring. There is no question that as the number of ICs increases, so does the probability of error.

With all things considered, the industry reports a nominal test cost of about $0.15 per IC.[27][28]

Test cost = $0.15 per 16 pin IC

REWORK

The cost of rework is best understood by considering the cause of errors in more detail. Errors are typically caused by poor board quality, inadequate solder process, tolerance of insertion, and of course, bad chips. Table 4 shows the average board fault spectrum. The figures are a conclusion reached by EVALUATION ENGINEERING magazine[10] as to what the industry is currently seeing. The table shows that the majority of board errors is due to solder shorts. These errors are the result of traces or IC holes being too close, which is what happens on densely populated boards.

Table 4. Average Board Fault Spectrum

Tolerance	20%
Shorts	40%
Insertion	30%
Bad Parts	10%

Of all the material costs associated with rework, the main cost is the time spent on a repair. Considering that it takes approximately two minutes to desolder, insert, resolder, and clean a component pin[9], one can see that more ICs on a board directly affect cost. Repair times also increase dramatically on multi-layer boards that might have been doubled sided if denser logic was used.

For our comparison, let's assume that our test equipment is 95% efficient in finding solder faults on the first pass (no loop). This leaves 5% of the faults that go undetected and eventually must be found and repaired. The estimated cost per pin based on a $6.00 hourly wage and the two minute repair time is approximately $0.02 cents.

Rework = [$0.02 × total pin count]/IC count

It is important to note that the probability of errors is based on a Poisson distribution[8] that increases exponentially with the number of pins and components. This distribution is used in wave solder processing to correct for solder errors. Mathematically this is expressed as:

$$P = \frac{e^{-np}(np)^x}{X!}$$

where; P = The probability that a defect will occur
n = The number of components
p = The fraction defective
x = The actual number of defects

This means that the TTL and PAL version of the arbiter have a higher probability of error than the EPLD version. However, to make our comparison easier, let's simplify this to more of a linear relation. For each implementation, the rework cost per IC is calculated by;

Rework cost = [(total pin count) × (5%) × ($0.02 cents)]/IC count

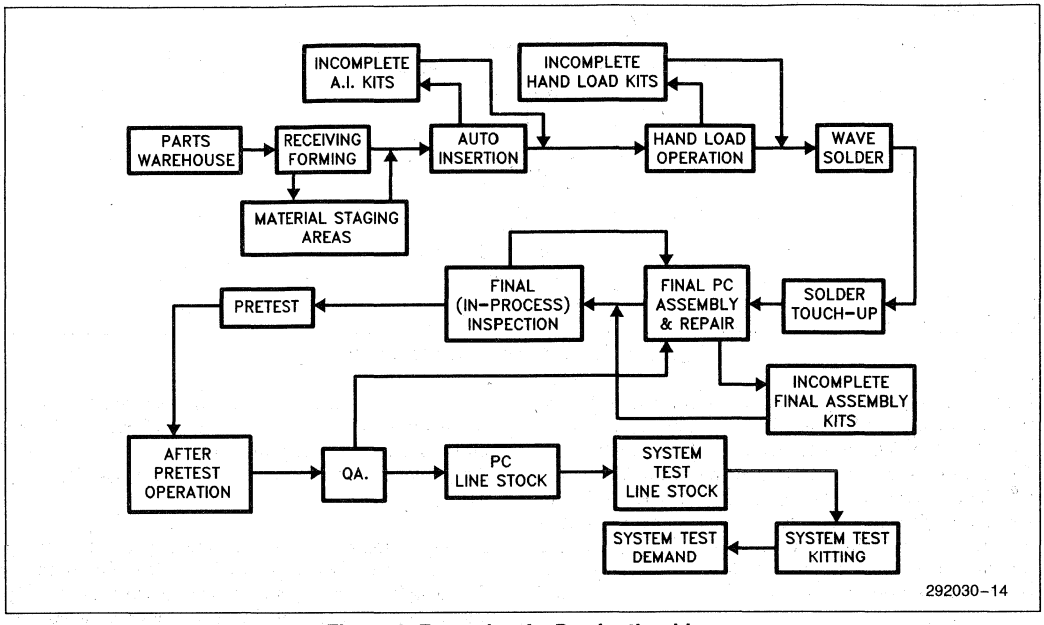


Figure 9. Example of a Production Line

QUALITY CONTROL

In most production operations, boards go through several steps of quality inspection. The bare board might be inspected after preliminary tests and after system tests. Although 100% inspection should theoretically eliminate all errors, in real life this rarely happens. The main reason for this is the complexity of the production and rework loops as shown in Figure 9.

Quality control's purpose is to remove defective products and either junk them or rework them, neither of which is cost effective. The best approach is to design the quality in, not fix it in. One way to design in quality is by reducing the possibility of errors and increasing the reliability of a product. This is one of the primary advantages of dense logic (like EPLDs and PALs) over TTL.

A survey conducted by *CIRCUITS MANUFACTURING* magazine[8] yielded the cost of $10 to $50 dollars to inspect, find, and repair a defect on a board. They summarized that the actual cost of inspection is about $0.004 for each hole on a board. With this in mind, let us assume a 100% inspection of our arbiter circuit for each implementation. This means that each pin (and every trace via) will have to be looked at. The calculation for this is;

$$\text{QC cost} = (\text{total pin count} \times \$0.004)/\text{IC count}$$

POWER SUPPLY

Price for 5V, single output, switching power supplies as advertised by several vendors is $1.00 per watt. The calculation for determining power supply costs in our comparison is:

$$\text{Power cost} = [(5\text{VDC} \times I_{CC}\ (\text{mA})) \times \$1.00\ \text{per watt}]/\text{IC count}$$

Additional Costs

In addition to the more obvious costs, there are several other items that contribute to the "hidden cost" of a system.

PROGRAMMING LOSS

Because PALs are a one time programmable type of device, full testing can't be done on them without destroying the user's fuses. For this reason PALs have a published programming loss of 2%[20]. The cost for this is:

$$\text{Programming loss} = (\text{PAL IC count} \times 0.02) \times \text{PAL cost per IC}$$

EPLDs, because they are based on EPROM cells, can be programmed for different patterns, fully tested before customer delivery, and then erased. The result is a near 100% percent programming yield[22].

PROGRAMMING FEE

Programming fee is the cost of programming a device. While many companies have in-house programmers, it is quite common for programming to be done by the distributor. In some cases, and at low volumes, the programming may be done free of charge. However, at larger volumes a programming charge is not uncommon. The charge varies with volume, programmer availability and in general, your state of affairs with the distributor. The cost for programming EPLDs and PALs is the same per device and averages about $0.25 cents.

Programming fee = $0.25 cents

SAFETY STOCK

Although this particular item was not mentioned in the inventory section, it plays a very important role in the production world. Safety stock[21] is extra ICs ordered to cover for unexpected events. Unexpected here might be a large unforeseen customer order or simply a bad batch of parts.

While industry seems to strive for the optimum JIT (just in time) production[14][16], which stresses minimal inventory until needed, it's not unusual for production managers to carry a five to ten percent inventory buffer depending on the cost of the part. In most cases, the larger expensive parts like microprocessors, peripheral controllers, and other LSI devices are safety stocked in smaller quantities.

Let's assume that the safety stock is to be a maximum of 10%. Five percent might be used to cover for the unexpected occurrences, and five for WIP (work in process) modifications. Since all parts have the same probability of unexpected events we can assign that percentage equally. Justifying the second 5% depends on the IC technology itself. For instance, WIP modifications usually require cuts and jumpers on TTL, therefore it's unnecessary to order the additional 5%. In process modifications to an EPLD are done simply by reprogramming it, here again there is no need for the additional 5%. PALs however cannot be cut and jumpered (internally) nor can they be reprogrammed. Also, there is the possbility that "on the shelf" PALs will be programmed in advance, therefore a WIP mod that impacts their function means that those parts must be obsoleted (junked). In this case, an additional 5% is justifiable.

Let us assume that the production manager reduces safety stock by a moderate amount, let's say 3%. In a case like this, usually the larger more expensive parts are curtailed first. Since EPLDs provide good coverage for work in progress and because they are more expensive by comparison, we can reduce the total safety stock to 2% and not compromise our safety margin. Because TTL is inexpensive it tends to suffer more of the "gun-shot" approach in testing[7]. This means that the useage rate is greater because production technicians tend to replace TTL parts with more liberty. For this reason let's leave the TTL safety stock as it stands. PALs could be reduced, but faced with the fact that the programming yield is 2% and that internal modifications can't be made, the production manager might decide not to change the safety stock for PALs. These results are shown in Table 5.

Table 5. Safety Stock

	TTL	PAL	EPLD
Unexpected Events	5%	5%	2%
WIP MODS	0	5%	0
Total	5%	10%	2%

The safety stock calculation for each implementation is:

Safety stock = (% of stock × IC type × IC type cost)/IC count

DE-COUPLING CAPACITORS

While adding caps solves many problems due to system noise, it also increases the cost of PCB layout, PCB fab, and adds an additional burden on all of our other costs. For a TTL system, a good de-coupling rule of thumb is to use one 0.01 μf per each synchronous driven gate and at least 0.1 μf per 20 gates regardless of synchronicity. Engineers recognize the need for decoupling and usually take it a step further by using one capacitor per IC. Most boards reflect this practice, which, in itself is very good. However, the addition of all these caps is definitely measurable, in both component and systems cost.

The average cost of a ceramic capacitor in moderate quantities is about half a cent. For our comparison we will follow the accepted practice and de-couple each TTL, PAL, and EPLD device. Our capacitor cost is then:

De-coupling cost = $0.005 × IC count

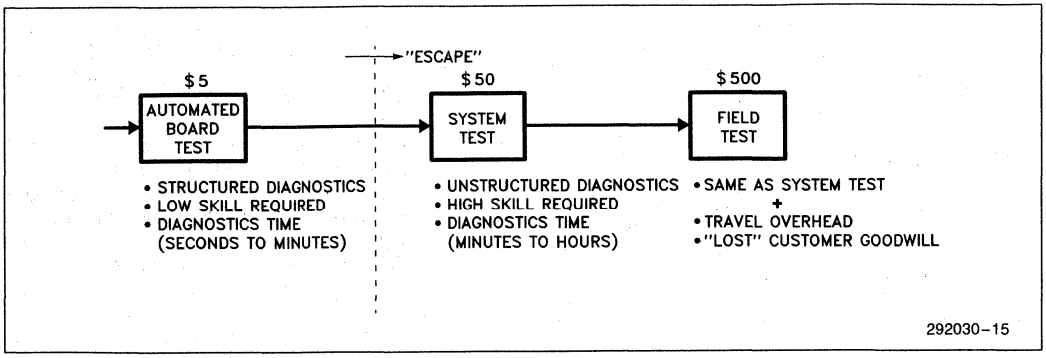


Figure 10. Escape Costs

Other Costs To Consider

Eventually, some place toward the end of a production line, a board becomes part of a system. At this point it is housed in an enclosure and all the necessary cabling is done. Even here, however, the impact of using a particular IC technology can still be felt.

DEFECT ESCAPES

One very significant item that the test community acknowledges is the cost of "escapes"[4]. "Escape" is defined as a fault that goes through the early stages of board test undetected. Figure 10 shows the escape relationship. An industry rule of thumb states that the cost to detect a fault increases by an order of magnitude at each stage. This means that if it costs $5 to find a fault at the board test level, that same fault might cost $50 at the system level and $500 at the field level. An important relationship to remember, is that the number of faults per board increases logarithmically, as the number of components on the board increases[6]. The cost of an "escape" is difficult to quantify, but generally, a board with a higher component count has a greater cost[2][8].

CABLES/WIRING HARNESS

When the number of components or the power requirements of a system are reduced, a reduction in cables and wiring is usually expected. The cost savings here is either in the elimination of cables (because more functions are condensed into an IC) or the reduction of cable gauge or length (because less power is required, in the case of EPLDs). Also, fewer cables means fewer cable ties, connector pins, and mounting hardware. While this is a subjective figure, lets assume that the distributed cost of system cables is $0.25 per IC.

$$\text{Cable cost} = \$0.25 \times \text{IC count}$$

ENCLOSURE

Certain applications require reduced packaging or enclosure size. In industrial control for example, each line might require a complete system to monitor it's operation. In a case like this, a large bulky box full of boards might not be appropriate. A good example of the benefits that high integration logic provide enclosures, is the third market versions of the popular PC. Many of these companies have fully compatible versions that fit on a single board. EPLDs and PALs are capable of providing a cost savings in this respect. However, while PALs approach the density requirements, their large power needs render them counterproductive to the low power specs of small systems. TTL is just not as effective as either PALs or EPLDs.

For our comparison let us assume the cost of enclosure per chip is $0.75. The calculation is:

$$\text{Enclosure cost} = \$0.75 \times \text{IC count}$$

Table 6 shows the cable and enclosure costs for the MULTIBUS I circuit. Although the results are based on assumed values, we can see that a larger IC count influences the burdened cost of the system. Our final comparison will not use these figures, but they should be considered.

Table 6. Other Production Costs for Multibus I Circuit

	TTL	PLA	EPLD
Wiring/harness	$2.750	$0.500	$0.250
Enclosure	$8.250	$1.500	$0.750

Arbiter Circuit Conclusion

A compilation of the cost variables for our comparison is shown in Table 7a and 7b. Because the cost may differ for each company, the comparison calculations were done on a Lotus 1-2-3 worksheet that the individual engineer can modify with their specific values. The worksheet is available, and can be downloaded from the Intel EPLD bulletin board. Table 8 shows our calculation results for three years of production.

Inventory:	Costs
Incoming insp. ($/pin)	$0.010
Storage ($/sq.ft./yr)	$20.000
Maintenance ($/part)	$0.030
Processing ($/part type/yr)	$0.520
Safety stock (%)	2%

Manufacturing:	Costs
PCB fab. ($/sq.in.)	$0.200
Assembly ($/part)	$0.100
Test ($/part)	$0.150
Rework ($/pin)	$0.020
QC ($/pin)	$0.004
Power ($/watt)	$1.000
Interconn	$0.020
Program ($/part)	$0.250
Caps. (each)	$0.005

(a)

Integrated Circuits

Component Count:

Package	TTL	PLA	EPLD
DIP14	10		
DIP16	1		
DIP20	0	2	
DIP24			1

ICs	Types
TTL	10
PLA	2
EPLD	1

Circuit Requirements:	I_{CC} (max)	Interconnects
TTL circuit (total mA).	100	36
PLA circuit (total mA).	240	7
EPLD circuit (total mA).	15	0

(b)

Tables 7a and b. Multibus Arbiter/Controller Cost Variables

Table 8. MULTIBUS I Arbiter/Controller Production Costs

AVERAGE COMPONENT COST									
	Year 1			Year 2			Year 3		
Package	TTL	PLA	EPLD	TTL	PLA	EPLD	TTL	PLA	EPLD
DIP14	$0.25			$0.20			$0.19		
DIP16	$0.35			$0.30			$0.27		
DIP20	$0.55	$2.00		$0.38	$1.70		$0.35	$1.56	
DIP24			$6.00			$4.20			$2.90
PRODUCTION COSTS									
	Year 1			Year 2			Year 3		
Item (costs per part)	TTL	PLA	EPLD	TTL	PLA	EPLD	TTL	PLA	EPLD
Components	$0.259	$2.000	$6.000	$0.209	$1.700	$4.200	$0.197	$1.560	$2.900
Incoming Insp.	$0.142	$0.200	$0.240	$0.142	$0.200	$0.240	$0.142	$0.200	$0.240
Inventory									
Maintenance	$0.027	$0.038	$0.045	$0.027	$0.038	$0.045	$0.027	$0.038	$0.045
Storage	$0.030	$0.042	$0.050	$0.030	$0.042	$0.050	$0.030	$0.042	$0.050
Processing	$0.473	$0.520	$0.520	$0.473	$0.520	$0.520	$0.473	$0.520	$0.520
Printed Circuit Board									
Fabrication	$0.043	$0.060	$0.072	$0.043	$0.060	$0.072	$0.043	$0.060	$0.072
Trace costs	$0.065	$0.070	$0.000	$0.065	$0.070	$0.000	$0.065	$0.070	$0.000
Assembly	$0.089	$0.125	$0.150	$0.089	$0.125	$0.150	$0.089	$0.125	$0.150
Board test	$0.150	$0.150	$0.150	$0.150	$0.150	$0.150	$0.150	$0.150	$0.150
Rework	$0.014	$0.020	$0.024	$0.014	$0.020	$0.024	$0.014	$0.020	$0.024
QC	$0.057	$0.080	$0.096	$0.057	$0.080	$0.096	$0.057	$0.080	$0.096
Power Supply	$0.045	$0.600	$0.075	$0.045	$0.600	$0.075	$0.045	$0.600	$0.075
Total Cost/Part	$1.393	$3.904	$7.422	$1.343	$3.604	$5.622	$1.331	$3.464	$4.322
Total Cost/System	$15.321	$7.808	$7.422	$14.771	$7.208	$5.622	$14.641	$6.928	$4.322
Additional Costs/System									
Programming loss	$0.000	$0.080	$0.000	$0.000	$0.068	$0.000	$0.000	$0.062	$0.000
Safety stock	$0.143	$0.400	$0.120	$0.115	$0.340	$0.084	$0.109	$0.312	$0.058
Programming fee	$0.000	$0.500	$0.250	$0.000	$0.500	$0.250	$0.000	$0.500	$0.250
De-coupling caps	$0.055	$0.010	$0.005	$0.055	$0.010	$0.005	$0.055	$0.010	$0.005
True mfg. cost/system	$15.518	$8.798	$7.797	$14.941	$8.126	$5.961	$14.804	$7.813	$4.635

The comparison in component costs shows that the EPLD costs more than either a TTL or PAL IC. As costs are added, the figures for TTL and PALs begin to approach the cost of an EPLD. These are shown on the line labeled "Total cost/part".

The "Total cost/system" line shows the actual cost when all the ICs are considered. For the first year, the TTL version is the more expensive implementation, and the EPLD numbers look very favorable.

The "True mfg. cost/system" line results after additional costs are figured in. Here we see that the first year, the EPLD version already provides a $1 savings over the PAL version, and that the cost of the TTL implementation is very high. Also, the inserted cost per IC at this point is, $1.15 for TTL, $2.40 for PAL and $1.80 for the EPLD. This is in line with the inserted costs that we mentioned earlier.

The production costs for two additional years shows that the decreasing price of EPLDs (based on the curve of Figure 5) will continue to provide costs savings as production ramps up in quantities.

In terms of functional benefits, the EPLD implementation is the most beneficial because;

- The chip count has gone down, one EPLD has replaced 11 TTL ICs in one implementation, and 2 PALS in the other, reducing the cost and time of:
  —board layout
  —board fab
  —assembly
  —rework
- The reliability of the board has increased. Fewer components translates into less probability of error.
- Modifications are easier to make. Instead of cuts and jumpers (for TTL), or throwing away a PAL, a change is re-programmed.
- The need for de-coupling caps is reduced. All those individual ICs are eliminated and in some cases the distributed capacitance of the board may be enough de-coupling.
- Power supply requirements are small. The active current requirements are much smaller with EPLDs. This in turn reduces the need for large power supplies and fans.
- Cable requirements and enclosure benefits have been improved. Since EPLDs provide better integration over TTL and PALs, the size of the system will be smaller. This translates into fewer boards and cables.
- Inventory is reduced. One EPLD replaces many TTL devices. Also, "on the shelf" programmed EPLDs can be reused in a pinch, PALs can't.

Less expense and probability of "escapes". The time and cost of finding and fixing escape problems is reduced to one reprogrammable IC. In the field, this translates into less "down time" for the customer and a higher level of customer "goodwill" for the OEM.

Allows capability for customized hardware. Specific customer requirements can be implemented. Also, DIP switches and configuration jumpers may not be necessary in many cases, since configurations can be programmed into the EPLD.

Development Costs

As mentioned earlier, the costs of development are usually dismissed as NRE. One reason for this is the difficulty in pegging down these costs. However, while money might be expendable at this stage, time is usually critical. Time saved at the front end can make a difference in beating the competition to market. The following topics are presented for consideration. No costs are assigned to them.

RESEARCH

The amount of time spent researching components, component sources, and technical data can be very large. Designs done with a large IC count require more research and analysis time. Higher integration devices require learning curve time, but, in the long run this tends to reduce research time, especially in future designs.

PROTOTYPING

For most companies, prototypes are three to five level wire wrap boards built by inhouse technicians or outside contractors. During prototype fab, a certain amount of work has to be done to each IC. Part of this work is, adding bypass caps, labeling chips, and lead forming. In smaller companies, the board might be hand wrapped. Larger companies might use an automatic wrapper. Once the board is wrapped, a continuity check is done on each wire net to insure connections and minimize shorts.

The turn around time for a protoboard is one to two weeks and can be shortened by paying a premium price. An alternate way of shortening this time is to simplify the board by using denser ICs.

DEBUGGING

Fixing bugs on a protoboard involves unwrapping and wrapping connections, as well as replacing ICs. Making mods on a TTL board is very time consuming and error prone due to the large numbers of wires. Making mods with PALs is expensive since the part usually has to be junked. EPLDs in contrast, are re-programmable and lend themselves to all the revisions that are common in the early design stages.

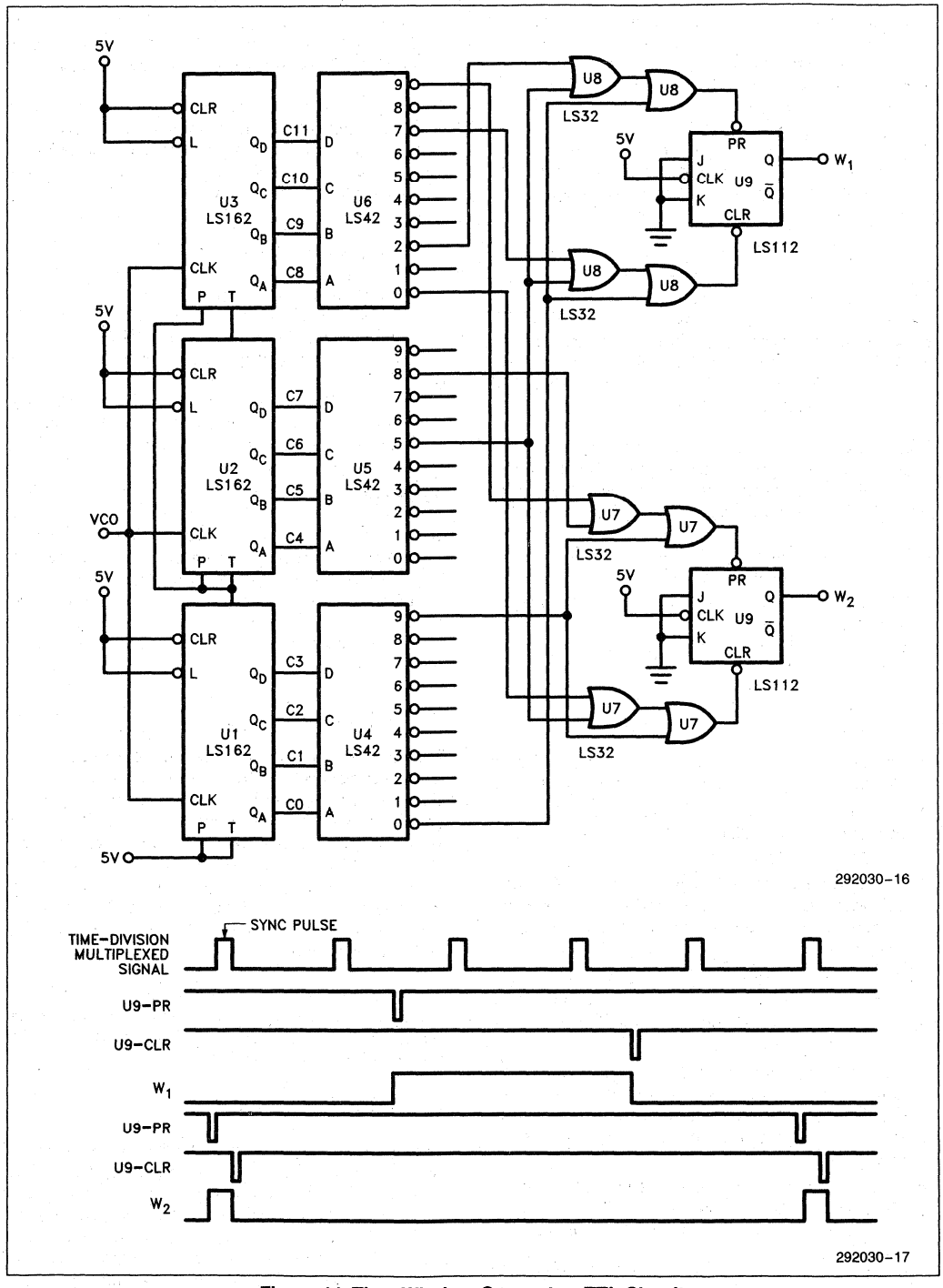


Figure 11. Time Window Generator, TTL Circuit

PCB LAYOUT

Artwork quotes are based on several factors. These are, board size, number of 16-pin chip equivalents, pad count, and the chip to board packing ratio. The chip equivalents are calculated by taking the total lead count (ICs and discretes) and dividing by 16. Pad count is the number of holes in the board. The packing ratio determines how much room an IC has around it. This is critical because space is needed to place sockets, vias, and trace bends. Currently, most service bureaus consider 0.75 square inches per IC to be the minimum packing density. This figure applies to DIPs only, other packages like SMT (Surface Mount Technology) will improve on this. However, for standard DIPs anything less than this might push the board into a multi-layer.

During schematic evaluation, the bureau doesn't usually charge for traces directly. Because they can't foresee the exact count, and they don't have time to count them on the sheets, they make a judgment based on previous jobs. If the board appears to be tight, their autorouter (CAD based) won't be as efficient, and more hand layout will have to be done. However, as more CAD based service bureaus integrate schematic capture front ends, the cost of traces and vias will be more visible.

Because the evaluation is subjective, the final cost varies, and is a combination of charges. However, because pad count can be determined easily, the overall price is usually gauged against a pad price.

WINDOW CIRCUIT

Background Information

In applications that involve time-division multiplexing, it is useful to have a circuit that windows a specific area of the bit stream [27]. The circuit of Figure 11 is a TTL implementation of such a circuit. The idea is to count time slots from a known reference and at a certain decode, set and clear a latch. The output of the latch is the time window, which might be used for further gating in other parts of the circuit. The TTL parts list is detailed in Table 9.

The PAL alternative of Figure 12 is comprised of two 16L8s and one 16R4. While the component count has been reduced from nine to three, there are still fourteen extra interconnections.

One 5C060 is needed to integrate the complete circuit. Fourteen out of the sixteen EPLD macrocells are used, and exernal traces are only the three I/O pins as shown in Figure 13.

Production Costs

The production variables for the window circuit are shown in Table 10a and 10b, and the production costs in Table 11. The comparison shows three years of system costs for each implementation.

Table 9. TTL Component List for Window Generator

IC	Type	DIP	I_{CC}(mA)	Area(In^2)	$
U1	LS162	16	32	.24	.49
U2	LS162	16	32	.24	.49
U3	LS162	16	32	.24	.49
U4	LS42	16	13	.24	.39
U5	LS42	16	13	.24	.39
U6	LS42	14	13	.24	.39
U7	LS32	14	9.8	.21	.18
U8	LS32	14	9.8	.21	.18
U9	LS112	14	6	.21	.29

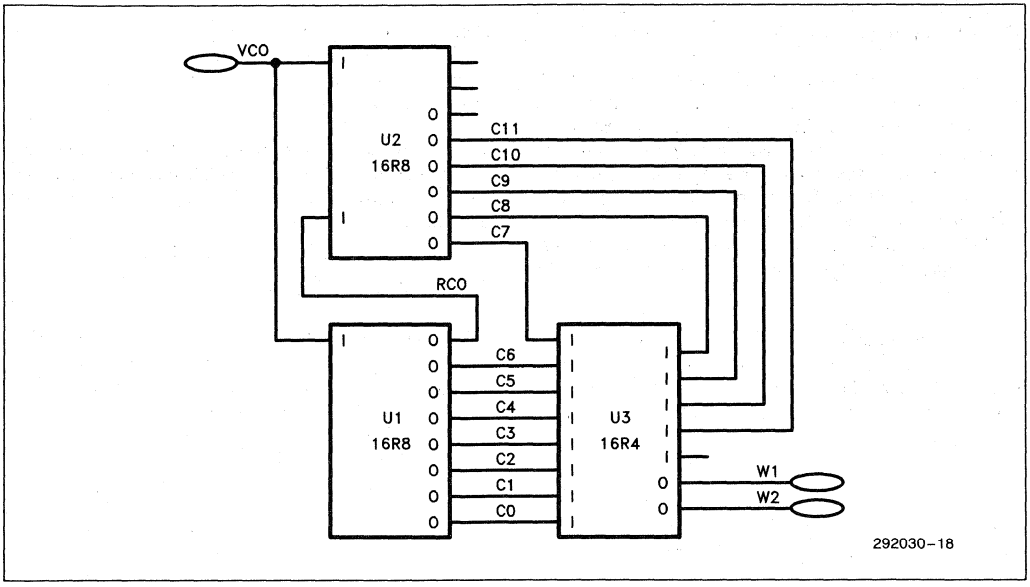


Figure 12. Time Window Generator, PAL Circuit

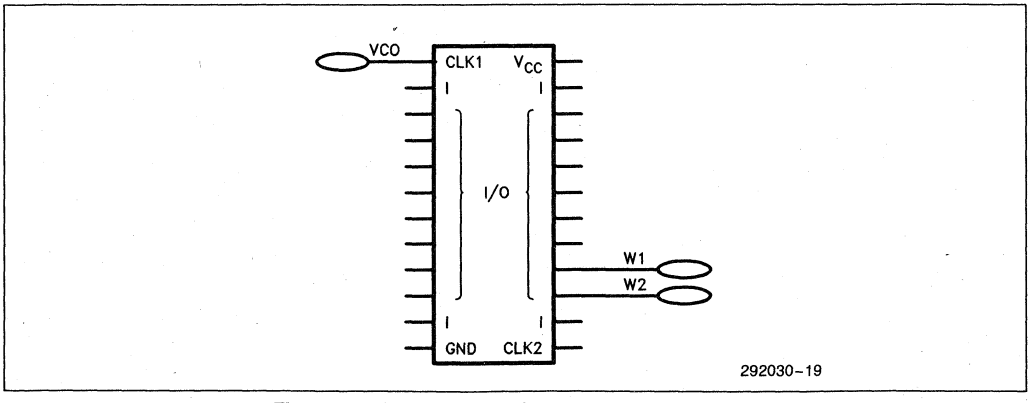


Figure 13. Time Window Generator, EPLD Circuit

Inventory:	Costs
Incoming insp. ($/pin)	$0.010
Storage ($/sq.ft./yr)	$20.000
Maintenance ($/part)	$0.030
Processing ($/part type/yr)	$0.520
Safety stock (%)	2%

Manufacturing:	Costs
PCB fab. ($/sq.in.)	$0.200
Assembly ($/part)	$0.100
Test ($/part)	$0.150
Rework ($/pin)	$0.020
QC ($/pin)	$0.004
Power ($/watt)	$1.000
Interconn	$0.020
Program ($/part)	$0.250
Caps. (each)	$0.005

(a)

Integrated Circuits

Component Count:

Package	TTL	PLA	EPLD
DIP14	3		
DIP16	6		
DIP20		3	
DIP24			1

ICs	Types
TTL	4
PLA	2
EPLD	1

Circuit Requirements:	I_{CC} (max)	Interconnects
TTL circuit (total mA).	160	52
PLA circuit (total mA).	360	14
EPLD circuit (total mA).	15	0

(b)

Tables 10a and b. Window Circuit Cost Variables

Table 11. Window Circuit Production Costs

AVERAGE COMPONENT COST									
	Year 1			Year 2			Year 3		
Package	**TTL**	**PLA**	**EPLD**	**TTL**	**PLA**	**EPLD**	**TTL**	**PLA**	**EPLD**
DIP14	$0.22			$0.19			$0.17		
DIP16	$0.44			$0.37			$0.26		
DIP20		$2.00			$1.70			$1.56	
DIP24			$6.00			$4.20			$2.90
PRODUCTION COSTS									
	Year 1			Year 2			Year 3		
Item (costs per part)	**TTL**	**PLA**	**EPLD**	**TTL**	**PLA**	**EPLD**	**TTL**	**PLA**	**EPLD**
Components	$0.367	$2.000	$6.000	$0.310	$1.700	$4.200	$0.230	$1.560	$2.900
Incoming Insp.	$0.153	$0.200	$0.240	$0.153	$0.200	$0.240	$0.153	$0.200	$0.240
Inventory									
Maintenance	$0.029	$0.038	$0.045	$0.029	$0.038	$0.045	$0.029	$0.038	$0.045
Storage	$0.032	$0.042	$0.050	$0.032	$0.042	$0.050	$0.032	$0.042	$0.050
Processing	$0.231	$0.347	$0.520	$0.231	$0.347	$0.520	$0.231	$0.347	$0.520
Printed Circuit Board									
Fabrication	$0.046	$0.060	$0.072	$0.046	$0.060	$0.072	$0.046	$0.060	$0.072
Trace costs	$0.116	$0.093	$0.000	$0.116	$0.093	$0.000	$0.116	$0.093	$0.000
Assembly	$0.096	$0.125	$0.150	$0.096	$0.125	$0.150	$0.096	$0.125	$0.150
Board test	$0.150	$0.150	$0.150	$0.150	$0.150	$0.150	$0.150	$0.150	$0.150
Rework	$0.015	$0.020	$0.024	$0.015	$0.020	$0.024	$0.015	$0.020	$0.024
QC	$0.061	$0.080	$0.096	$0.061	$0.080	$0.096	$0.061	$0.080	$0.096
Power Supply	$0.089	$0.600	$0.075	$0.089	$0.600	$0.075	$0.089	$0.600	$0.075
Total Cost/Part	$1.385	$3.754	$7.422	$1.328	$3.454	$5.622	$1.248	$3.314	$4.322
Total Cost/System	$12.463	$11.263	$7.422	$11.953	$10.363	$5.622	$11.233	$9.943	$4.322
Additional Costs/System									
Programming loss	$0.000	$0.120	$0.000	$0.000	$0.102	$0.000	$0.000	$0.094	$0.000
Safety stock	$0.165	$0.600	$0.120	$0.140	$0.510	$0.084	$0.104	$0.468	$0.058
Programming fee	$0.000	$0.750	$0.250	$0.000	$0.750	$0.250	$0.000	$0.750	$0.250
De-coupling caps	$0.045	$0.015	$0.005	$0.045	$0.015	$0.005	$0.045	$0.015	$0.005
True mfg. cost/system	$12.673	$12.748	$7.797	$12.137	$11.740	$5.961	$11.381	$11.269	$4.635

The production costs again show that the system cost for the first year is better with EPLDs. The two consecutive years show that the declining price of EPLDs make them an excellent candidate for systems that will ramp up production at that time.

Window Circuit Conclusion

The TTL version of the circuit was implemented with MSI counters and decoders. As a result, the PAL implementation was bound by the number of count bits and had to be programmed into two PALs. In circuits like this, it is useful to rewire the decode for different counts depending on the application. The PAL implementation allows this by incorporating the decode and output latches into one IC.

The EPLD implementation tackles the MSI integration quite easily and also provides the capability to reprogram the decoder. Since the counter and output latches consist of fourteen registered outputs, the sixteen macrocells of the 5C060 easily accommodate the needed functions.

SUMMARY

We have examined the hidden costs of production and how they differ for several logic alternatives. By examining these costs, we have shown that while an EPLD is presently a more expensive part, it's level of integration reduces system costs and improves reliability. The following items should be considered when evaluating logic alternatives:

- system cost is determined by more than component cost
- system cost and reliability is influenced by the type and amount of components used
- semiconductors have a life cycle that determines their present price at design, and at production time

In summary, when all system costs are considered, EPLDs can provide cost savings to the design and production of most board designs.

REFERENCES

1. The Future Is Now: Extending CAE into test of custom VLSI.
Robert S. Broughton, Tektronix.
Michael G. Brashler, Tektronix.
IEEE International Test Conference Proceedings, 1984

2. Reducing The Cost of Quality Through Test Data Managment.
Paul N. Manikas, GenRad Inc.
Stephen G. Eichenlaub, Harvard University.
IEEE International Test Conference Proceedings, 1983

3. A Quantitative Analysis Of The Trade-offs Between Higher Capital Investment and Higher Yield In PCB Testing.
Mark A. Myers, Teradyne Inc.
IEEE International Test Conference Proceedings, 1984

4. An Analysis Of The Cost And Quality Impact Of LSI/VLSI Technology On PCB Test Strategies.
Mark A. Myers, Teradyne Inc.
IEEE International Test Conference Proceedings, 1983

5. IC Quality Control By The User.
Roger Dunn, Xerox Corp.
IEEE International Test Conference Proceedings, 1983

6. An Analysis Of The Economics of Self Test.
P. Varma, University of Manchester.
A. P. Ambler, University of Manchester.
K. Baker, GEC Research Labs.
IEEE International Test Conference Proceedings, 1984

7. In Circuit Testability Factors: Shoot With A Rifle.
Douglas W. Raymond, Zehntel Production Services.
IEEE International Test Conference Proceedings, 1984

8. Seven Steps To Zero Defects.
D. W. Rudd, AT&T Technologies.
Circuits Manufacturing, June 1986

9. Rework Forum
Donald Ford, Senior Editor.
Circuits Manufacturing, September 1986

10. Manufacturing Defect Analyzers: Annual Round-up.
Evaluation Engineering magazine, August 1986

11. Assembly: Automation Makes It Better.
Roland W. Roy and Gordon Weeks, Andover Controls.
Circuits Manufacturing, February 1986

12. Shrinking Lines Squeeze Processes.
Jerry Murray, West Coast Editor.
Circuits Manufacturing, September 1986

13. Ribbon Cable for Reliable Interconnections.
Bennett W. Brachman, Xport Trading Inc.
Electronic Packaging and Production magazine, July 1986

14. TQC and JIT: Partners In Production.
Rick Walleigh, Hewlett Packard.
Circuits Manufacturing, February 1986

15. Automated Handling/Sorting: Multisite Development Moves to Back Burner.
Evaluation Engineering magazine, May 1986

16. Software Charts The Course of Component Testing.
Ronald Pound, Editor.
Electronic Packaging and Production magazine, June 1986

17. Complexity, PLDs Drive The Market.
Evaluation Engineering magazine, July 1986

18. Intel User Defined Logic Handbook.
Intel Corp. 1986

19. VLSI Semicustom Design Guide.
CMP Publications, Summer 1986

20. AMD Programmable Array Logic Handbook.
Advanced Micro Devices, 1984

21. Handbook Of Industrial Engineering.
Gavriel Salvendy, Editor, Purdue University
John Wiley & Sons Publications

22. Components Quality/Reliability Handbook.
Intel Corporation.

23. The Cost Edge.
DM DATA Corp.
Scottsdale, AZ

24. Semiconductor Purchasing Strategies Integrated Circuits Engineering Corp.
Scottsdale, AZ

25. Status 1986 Integrated Circuits Engineering Corp.
Scottsdale, AZ

26. EDN Semicustom Design Series
EDN Magazine, 1985

27. EDN Design Ideas
EDN Magazine, 1985

June 1990

Metastability Characteristics of Intel EPLDs

THOM BOWNS
PROGRAMMABLE LOGIC APPLICATIONS
INTEL CORPORATION

Order Number: 292071-001

Metastability Characteristics of Intel EPLDS

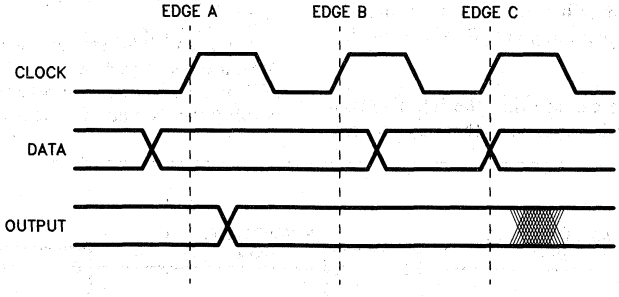
WHAT METASTABILITY IS

Edge triggered registers (otherwise known as "flip-flops") are designed to propagate and store data applied to their "D" inputs. If the data applied is HIGH just before the clock edge, the Q output transitions to HIGH. If the data applied is LOW just before the clock edge, the Q output transitions to LOW. These are the two normal states for a register. However, registers may be forced into a "metastable" state, in which the output either oscillates or simply remains in between a HIGH and a LOW for a prolonged period. This can occur when the register's setup or hold parameters are violated; that is, when data input to the register transitions too close to the clock edge. Figure 1 shows the three possibilities of when data may arrive with reference to the system clock, and when metastability may occur.

HOW METASTABILITY AFFECTS SYSTEM DESIGN

Most systems have the capability to read input data from an asynchronous source such as a keyboard or a modem. They are called asynchronous because the timing of the data from these sources is not in synchronization with the system clock. This type of data must first be synchronized before the system can use it; this usually means placing a synchronization register stage between the asynchronous data source and the rest of the system. Because the data is asynchronous, it inevitably violates setup and hold requirements of the synchronization register at some time during normal operation and drives the register into a metastable state. Once in the metastable state, the synchronization register eventually resolves to either a HIGH or LOW. How easy the register can be driven into, and how long it remains in the metastable state depends upon noise, ambient conditions, and the register's process technology. For example, FAST* TTL is more immune to and resolves more quickly from a metastable state than LS TTL.

In some older technologies, register outputs are fed directly to the device pin with no intermediate amplification stage. When one of these registers enters a metastable state, the voltage at the output pin can be seen to oscillate or hover in between HIGH and LOW. Intel CMOS EPLDs have high gain buffers between the register circuitry and the output pin. The output pins will always be at one rail or the other, and therefore will never show an in-between or oscillatory state. Instead, they manifest metastability by a late transition: when metastable, the output transitions appreciably later than allowed by the normal clock to output delay (T_{CO}) specification of the device. To guard against the effects of metastability, one designs the system to wait a sufficient time after the register's specified T_{CO} to ensure valid data from the register. This additional wait time after T_{CO} is called T_{MET}. The designer uses the device's supplied metastability characteristics, chooses a Mean Time Between Failures (MTBF) value that meets the target system requirements, and then calculates a T_{MET} based upon that MTBF.



Edge A: Data changes before clock; normal output transition
Edge B: Data changes after clock; no output transition
Edge C: Data changes with clock; output goes metastable

Figure 1. Synchronization Data vs. Clock Timing

*FAST is a trademark of Fairchild Semiconductor Corporation.

Since metastability is a probabilistic quantity, one cannot simply say that after a given T_{MET} the register is absolutely guaranteed to have resolved, but rather we refer to the probability that the register will have resolved. Further, since MTBF is exponential in nature, the failure curve is greatly skewed toward early failures. For instance, say a given register claims an MTBF of 100 years for a given T_{MET}. Intuitively, we would expect a 50% probability of a failure in the first 100 years. However, the probability is actually 63.2% that the register will manifest a resolution failure in the first 100 years. Equation 1 gives a very close approximation of the probability of failure based on the MTBF of each synchronizer, the number of systems in use, the number of synchronizers on each system, the length of time in service, and the number of failures allowable in that time. As an example, given ten systems that each have five synchronizers with MTBFs of 1000 years, what is the possiblity of one failure in five years? Using equation 1 and solving for P (Probability):

Equation 1

$$P = 1 - (e^{(-T/MTBF)} ** (U/F))$$

$$P = 1 - (e^{(-5/1000)} ** (50/1))$$

$$P = 1 - (e^{(-0.005)} ** 50)$$

$$P = 1 - (0.995 ** 50)$$

$$P = 1 - (0.775)$$

$$P = 0.221 = 22.1\%$$

Where T = time before failure in years, U = number of synchronizer units operating, and F = the number of failures expected.

From this equation we can see that the intuitive concept of MTBF is inadequate, and that an accurate failure prediction algorithm is necessary. The concept of MTBF is very complicated and may seem unnecessarily technical, but it is essential to know what you're getting when a PLD manufacturer claims 100 years MTBF; that's for one synchronizer alone. Adding more synchronizers increases the probability of failure that much more.

Figure 2 shows the simplified diagram of a synchronization register feeding the rest of the system, which is shown here as simply another register. The minimum clock period for a single stage synchronization scheme such as this is given by equation 2. The clock period cannot be any shorter than the maximum TCO of the synchronization register plus the maximum TSU of the system plus TMET for the given system conditions.

Equation 2 solves for the margin afforded by a given clock frequency.

Equation 2

$$\text{Margin} = 1/F_C - (T_{SU} + T_{CO} + T_{MET})$$

where F_C is the system clock frequency, T_{SU} is the system's input data setup to clock edge minimum time, T_{CO} is the synchronization regsister's clock to output delay, and T_{MET} is the additional time after T_{CO} that the designer is willing to wait. Equation 3 allows the designer to determine what sort of T_{MET} he can expect for a chosen MTBF.

Equation 3

$$T_{MET} = (\tau \bullet \ln(T_W \bullet F_C \bullet F_D \bullet MTBF)) - T_{CO}$$

where τ and TW are the metastability characteristic constants for the selected synchronization register type, MTBF is the mean time between failures expected, F_C is the system's clock frequency, and F_D is the average

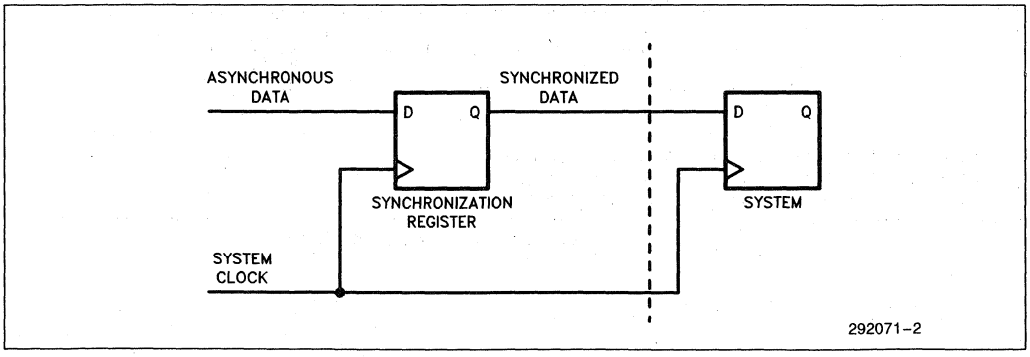


Figure 2. Single-Stage Synchronizer

data rate coming into the synchronization register. Equation 4 was derived from equation 3 to solve for MTBF when T_{MET} is chosen and the other factors are known.

Equation 4

$$MTBF = \frac{e^{\left(\frac{T_{MET} + T_{CO}}{\tau}\right)}}{T_W \bullet F_C \bullet F_D}$$

It is important to note that each register within an Intel EPLD is independent; if one register should go metastable, other registers within the device are not affected. Thus if several macrocell or input registers are used in parallel as a byte-wise synchronizer array, one need not be concerned about one register affecting the data in another register. The only problem that may arise would be if the incoming data bytes had some timing skew between the bits that make it up. If one input data bit transitioned later than the others, driving it's synchronizer into metastability, the worst case scenario would be that the system would sample the synchronizers before that register settled. That would mean that the data at that point would most likely be invalid, and the next sampled data byte might also be invalid. However, with a sufficient T_{MET} to allow for metastable settling, the probability of this occurrence can be reduced to a managable level.

HOW METASTABILITY IS QUANTIFIED

Each device has two metastability characteristic constants, τ (or "Tau") and T_W, which are used to determine the probability of metastable occurrences. As shown in Figure 3, this probability decreases at an exponential rate, according to the value of τ. T_W defines the likelihood that the register will enter the metastable state in the first place; it is also known as the "failure window". Given τ and T_W, T_{MET} can be calculated for a chosen MTBF, according to equation 3.

The metastability constants are determined by running the register under test through billions of clock cycles with randomly changing input data. For each increment of T_{MET}, the number of late transitions are counted within a given time period. A late transition is when the delay from the clock edge until when the register's output changes takes longer than it's observed T_{CO} allows; we interpret that as an occurrence of a metastable event. The number of failures recorded for each increment of T_{MET} is logged and then run through the following equations, producing τ and T_W.

3

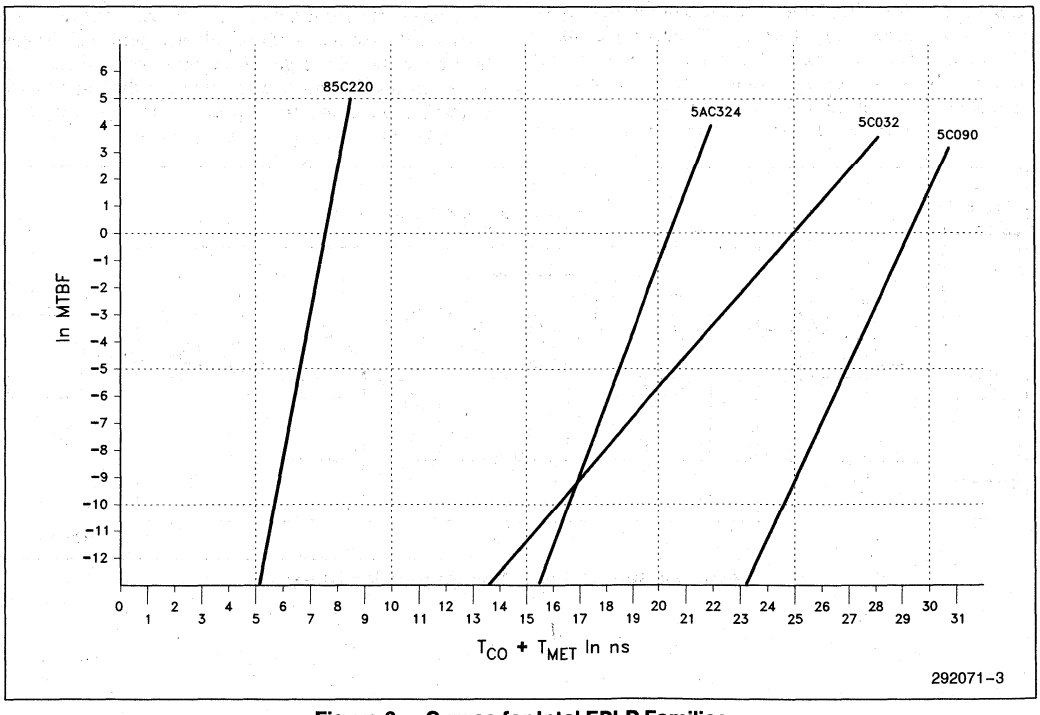


Figure 3. τ Curves for Intel EPLD Families

Equation 5

$$\tau = \frac{n \sum_{i=1}^{n} Xi^2 - \left(\sum_{i=1}^{n} Xi\right)^2}{n \sum_{i=1}^{n} Xi\,Yi - \left(\sum_{i=1}^{n} Xi\right)\left(\sum_{i=1}^{n} Yi\right)}$$

Equation 6

$$T_W = \frac{\sum_{i=1}^{n} \left(\frac{N_i \bullet e^{\left(\frac{T_{CO} + T_{MET}}{\tau}\right)}}{Nc_i \bullet F_C} \right)}{n}$$

The X axis is scaled as the $T_{CO} + T_{MET}$ value in nanoseconds, and the Y axis is the Natural logarithm of the number of seconds between failures (actual number of failures recorded divided into the amount of time taken to accumulate those failures). X_i and Y_i are the delay value and log of the failures, respectively, at point i. N_i is the number of events recorded at point i, and N_{ci} is the number of clock pulses which occurred within that time.

When plotted on semi-log paper, the resulting line shows the τ slope. The Y-axis intercept point is related to T_W.

Metastable failure characteristics for each EPLD are ascertained by using the setup in Figure 4. The register is clocked by a 2MHz, 25% duty cycle pulse, and register data is provided via a 1MHz, 50% duty cycle pulse. The data is free running with respect to the clock to afford truly random data distribution with the base period of about 1 transition per clock cycle. Since the data transitions are evenly distributed over the entire clock period, we may assume that the data transitions are also evenly distributed within the failure window (violating setup and hold). The register under test or DUT (device under test) clock is provided by the first output of a dual pulse generator. The second output is delayed by $T_{CO} + T_{MET}$ and fed into the metastable characterization unit. The output of the DUT is thus sampled at $T_{CO} + T_{MET}$, and it is sampled again aproximately 100 ns later. If the two samples disagree, the DUT must have transitioned after $T_{CO} + T_{MET}$, and therefore must have been metastable. For each metastable event detected, a pulse is generated to increment the event counter. In about half of the cases, the DUT may enter the metastable state and resolve to the same logic level as before, and so not give an external indication of metastability. However, since cases like this will not affect system performance, they are not figured into the MTBF equations.

To run the test, T_{CO} is characterized for ambient conditions and used as a base delay. The first test begins, and the counter catches X metastable failure events for 60 seconds. The number of metastable failures is recorded for that particular T_{MET}. T_{MET} is increased by .2 nS and another X events are counted for 60 seconds. This is continued until the events get spaced out to where less than 100 events occur in 60 seconds, and the remaining data points are arrived at by counting 100 events for X seconds and recording the results appropriately. The results are plotted on semi-log paper, and the slope of the resultant curve is τ . Since the raw data points will most likely not fall along a perfectly straight line, the only way to get an accurate value for τ is to either draw a best-fit line through the data points and physically measure the slope, or to run the data through equations 5 and 6, given above.

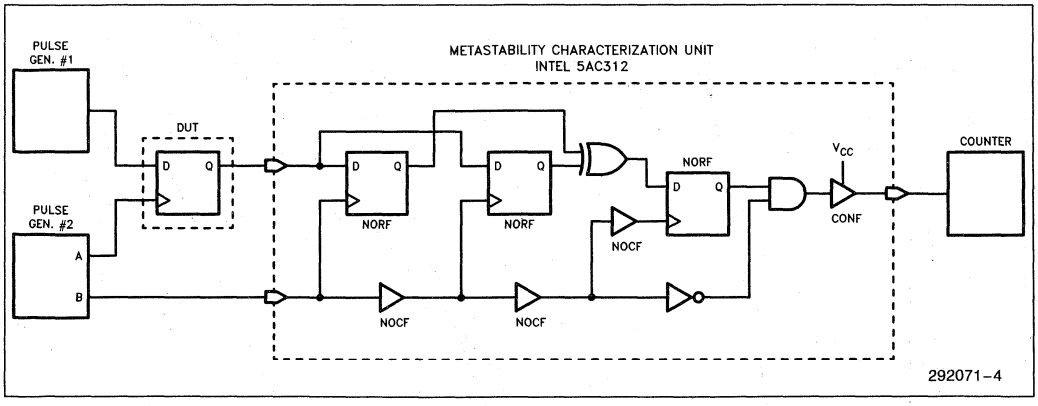


Figure 4. Metastability Characterization Setup

METASTABILITY DATA FOR INTEL EPLDS

Intel EPLDs are based upon fast, CMOS EPROM process technologies which make them inherently resilient against metastability. The metastability characteristics for each device are dependent upon its design and process technology, thus all Intel EPLDs within a given family have similar metastable characteristics.

The metastability characteristic constants for each Intel EPLD are shown in Table 1. Corresponding metastability characteristic curves for each family are shown in Figure 3.

HOW INTEL EPLDS COMPARE AGAINST OTHER PLDS

The first half of Table 2 lists maximum allowable clock frequencies for a single-stage synchronization scheme for various EPLDs. This determination comes from a derivation of equations 2 and 3, with the margin set for 0 nanosecond. The other factors are calculated to give a resulting maximum clock frequency. The second half of Table 2 shows the T_{MET} values that achieve different MTBF levels for several PLD technologies. The metastability characteristics of the other PLDs were determined empirically using the same bench testing method as the Intel EPLDs. It is easy to see that Intel EPLDs have very good metastable resolution characteristics when compared against other technologies.

Table 1. Intel EPLD Metastability Characteristic Constants

Device	τ	T_W
5C031	0.82 ns	5.8×10^{2}
5C032	0.86 ns	2.4×10^{-2}
5C060	0.90 ns	4.0×10^{0}
5C090	0.44 ns	2.0×10^{14}
5AC312	0.42 ns	1.4×10^{4}
5AC324	0.35 ns	8.6×10^{10}
5AC324 (Input Registers)	0.39 ns	3.0×10^{19}
85C220	0.22 ns	5.2×10^{1}

Table 2. Maximum Clock Frequencies for Sampled PLDs

Margin = 0 ns FD = 4 MHz	**Maximum Clock Frequency for a Chosen MTBF** (Based on Equations 2 and 3)				
	Device				
MTBF	**5C032-30**	**5AC312-25**	**85C220-80**	**16V8A-10**	**16R6-7**
1 Yr	16 MHz	22 MHz	53 MHz	46 MHz	26 MHz
10 Yrs	15 MHz	22 MHz	51 MHz	45 MHz	24 MHz
100 Yrs	15 MHz	21 MHz	50 MHz	44 MHz	23 MHz
1000 Yrs	15 MHz	21 MHz	49 MHz	43 MHz	21 MHz

FC = 33 MHz FD = 4 MHz	**MTBFs for a Given T_{MET}** (Based on Equation 3)			
	Device			
MTBF	**5C032-30**	**85C220-80**	**16V8A-10**	**16R6-7**
1 Yr	22.1 ns	6.3 ns	3.9 ns	25.1 ns
10 Yrs	24.1 ns	6.8 ns	4.3 ns	27.7 ns
100 Yrs	26.1 ns	7.3 ns	4.6 ns	30.2 ns
1000 Yrs	28.0 ns	7.8 ns	5.0 ns	32.7 ns

SUMMARY

Every dynamic system which has two stable states can enter a metastable state. Each PLD technology has a metastable characteristic associated with it, and some are less susceptible to metastability than others. By using Intel CMOS EPLDs and knowing how to effectively guard against the effects of metastability, systems may be designed to be virtually trouble-free.

REFERENCES

1. Mark Johnson, MIPS Computer Systems, "Arbiter/ Synchronizer Failure", COMP.ARCH, USENET Public Domain Computer Network, September 1989
2. Charles Dike, Signetics, *A Metastability Primer*, January 1989
3. Thomas Chaney, "Measured Flip Flop Responses To Marginal Triggering", *IEEE Transactions on Computers*, Volume C-32, No 12, December 1983
4. Peter Stoll, *How To Avoid Synchronization Problems*, VLSI Design, November/December 1982

ACKNOWLEDGEMENTS

Thanks to members of Intel's engineering staff, especially to Mike Allen and Duane Chinnow for assistance with the conceptual background for this application note.

September 1990

3

PLD Quality and Reliability Data Summary

Order Number: 293003-002

PLD QUALITY AND RELIABILITY DATA SUMMARY

CONTENTS

CONTENTS

THE IMPORTANCE OF RELIABILITY

Reliabilty of the erasable programmable logic devices (EPLDs) in your end product is critical to your total system reliability. The use of Intel EPLDs can make a difference. Intel EPLDs are manufactured on patented EPROM processes with proven reliability.

Quality ≠ Reliability

A quality component is one that meets your specification when received and tested. A reliable component continues to meet your specification even years after you have shipped your product. While Intel is a quality leader, we also adhere to stringent reliability standards which we have established for ourselves.

Consider Quality vs Reliability

The true cost of any component involves more than just the purchase price. The true component cost encompasses the initial purchase price, cost of rework during system production, and the cost of field repairs due to component failures. "Rework" costs during system production are incurred prior to shipment of your end product, and are a function of the quality of the component you purchase.

Repair costs incurred in the field after end product shipments, are a function of the reliability of the components. In addition to the increasing real cost of a system field service call, there is the intangible cost of a poor reliability reputation to the end user of your product. These costs depend upon the reliability of the components you purchase. Thus, reliability may impact costs during the system lifetime more than the initial quality of the components!

The Roots of Reliability

The manufacture of a reliable semiconductor device using a modern technology is a dynamic and evolutionary process. Success of this process is highly dependent upon the interplay between knowledgable and experienced manufacturing engineers, materials physicists, and responsible/responsive management. Only the correct combination can consistantly deliver high volumes of a reliable product. In this model, the experienced process engineer selects and defines the stresses to be performed and the performance criteria to be met, utilizing appropriate statistical tools and limits. The materials physicist then determines the root cause of the failure, if and when failure occurs, and provides effective solutions and/or containment recommendations. Finally, management provides the resources for the entire process from initial monitor to root cause corrective action.

Monitor Program

Reliability is designed into each component Intel manufactures. From the moment the design is put to paper, stringent reliability standards must be met at each step for a product to bear the Intel name.

Designing-in reliability, however, is only the beginning. Ongoing tests must be conducted to ensure that the original reliability specifications remain as valid in volume production as they were when the device was first qualified.

Intel's Reliability Monitor Program, devised to measure and control device reliability in production, is a proven tool that Intel has used for seven years and is now available to its customers. The Monitor Program subjects all of Intel's technologies to a 48 hour dynamic burn-in at 125°C (with a portion of these devices continued for a 1000 hour lifetest) and provides answers about device reliability that are not generally available from limited testing programs. But it's much more than burn-in and device testing. When test rejects are encountered, failure analysis is performed on each failed part. Isolating the fault and determining the failure mechanism is a critical part of the Monitor Program. It is the most comprehensive reliability program anywhere.

The paramount objective is to deliver reliable, quality devices. Actions that Intel takes to meet this objective may include a process or design change, or added reliability screen. Each decision is made with our customers in mind so that they receive the parts–and the performance–that they ordered by specifying Intel. Reliability qualification assures that all new production meets Intel's reliability standards. The Reliability Monitor Program ensures that these high standards are continually maintained, day in, day out, over the duration of a device's life. This reliability improves the life-time reputation of your product, reducing the required number of field service calls.

EPLD RELIABILITY DATA SUMMARY

Intel routinely publishes this "EPLD Reliability Data Summary", a continuing update of reliability information covering Intel's EPLD product line. This document includes a discussion on EPLD reliability testing methodology and the most current failure rate calculations, failure analysis and lifetest results. The "EPROM Reliability Data Summary" (order number: 210473 and 293004) should be used as a supplement to monitor EPROM process reliability.

Intel's commitment to the reliability of our products is clearly reflected in the information we make available to customers. We believe that supplying detailed reliability information to our customers is part of the total solution Intel offers, and is an important part of Intel's leadership in microelectronics technology.

EPLD Reliability Testing

Intel EPLDs undergo comprehensive testing to insure electrical reliability. This testing is done at qualification and/or during ongoing monitor checks. Testing differences between plastic and ceramic packaged EPLDs are noted.

Intel continually reviews its testing procedures and makes improvements to its methodology whenever overall reliability can be enhanced. Our goal is to be the industry leader in delivering high-quality, reliable parts.

Information on Intel's reliability testing procedures follows.

High Temperature 5.25V Dynamic Lifetest—This test is used to accelerate failure mechanisms by operating the devices at an elevated temperature of 125°C and nominal V_{CC}. During the test the device is programmed as a counter, with the inputs and outputs exercised, but not monitored or loaded. A pattern with greater than 90% of the EPROM cells programmed is used to simulate random customer patterns expected during actual use. Results of the lifetest have been summarized along with the failure analysis. Table 1 lists the activation energies for various failure mechanisms. These activation energies are used to calculate the amount of acceleration due to increased stress temperature or voltage (see Appendix A for failure rate calculations).

In order to best determine long-term failure rate, all devices used for lifetesting are first subjected to standard INTEL testing. The 48 hour burn-in results are an indication of infant mortality. These results are not included in the failure rate calculation. (See Figure 1 for typical lifetest bias and time diagrams.)

Table 1. Failure Mechanism Activation Energies

Failure Mechanism	E_a (ev)
Oxide	0.3
Fab/Assembly Defect	0.5
SBCL/SBCG/MBCL/MBCG	0.6
Contamination	1.0
Speed Degradation	0.3–1.0
Intrinsic Charge Loss	1.4

Failure Definitions

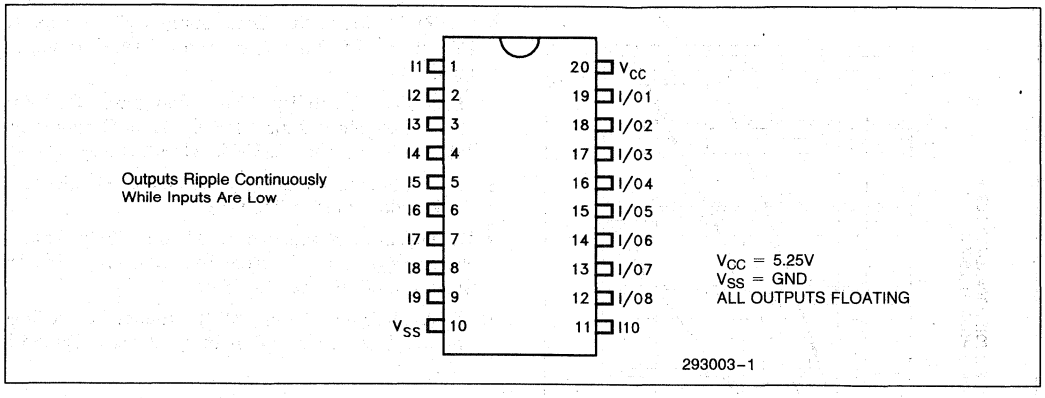
Oxide—An Oxide Failure Related Fault
SBCL—Single Bit Charge Loss
SBCG—Single Bit Charge Gain
MBCL—Multiple Bit Charge Loss
MBCG—Multiple Bit Charge Gain
Contamination—Ionic Contamination Failure
Speed Degradation—Device Speed Degraded Over Test

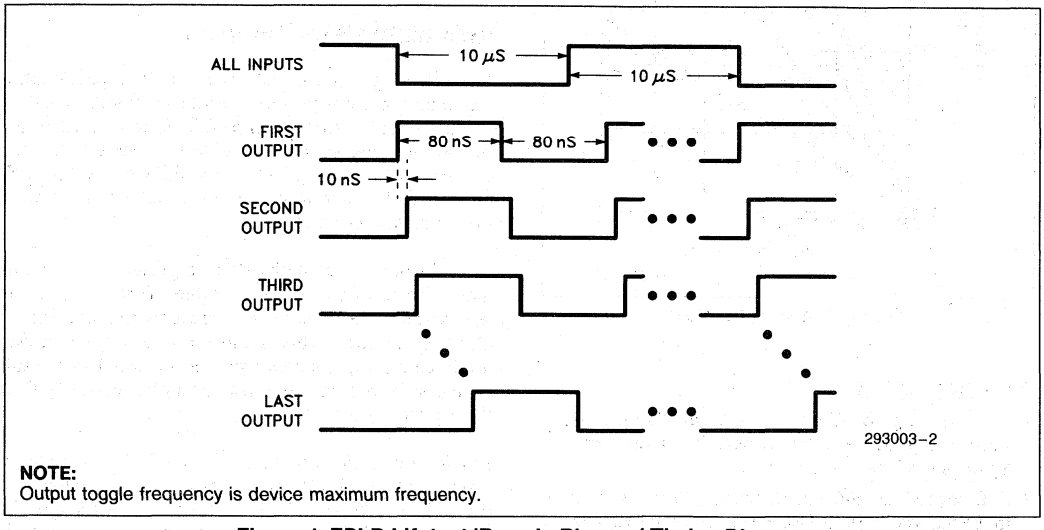
Failure Rate Calculations—Failure rate calculations are given for each relevant activation energy. Failure rate calculations are made using the appropriate energy (1, 2, 3, 4) and the Arrhenius Plot as shown in Figure 2. The total equivalent device hours at a given temperature can be determined. The failure rate is then calculated by dividing the number of failures by the equivalent device hours and is expressed as a %/1000 hours. To arrive at a confidence level associated failure rate, the failure rate is adjusted by a factor related to the number of device hours using a chi-square distribution. A conservative estimate of the failure rate is obtained by including zero failures at 0.3 eV. Devices submitted to stresses other than lifetest received a 168 hour lifetest prior to stressing. Appendix A provides example failure rate calculations.

High Temperature, High Voltage Dynamic Lifetest—This test is used to accelerate oxide breakdown failures. The test setup is identical to the one used for the dynamic lifetest except V_{CC} is an elevated voltage (6.50V for CHMOS IIE and 7.0V for CHMOS IIIE). The acceleration factor due to this test can be found in Appendix A. This data plus the standard dynamic lifetest data are used to calculate the 0.3 eV failure rate.

High Temperature Storage—This test is used to accelerate charge loss/gain from the EPROM floating gate. The test is performed by subjecting devices containing a 90%+ programmed pattern to a 250°C bake (140°C for plastic) with no applied bias. In addition to EPROM cell integrity, this test is used to detect mechanical reliability problems (e.g., bond strength) and process stability. This test is sometimes referred to as Data Retention Bake Test.

Temperature Cycle—This test consists of cycling the temperature of the chamber housing the subject devices from −65°C to +150°C and back. The device is functionally tested before and after the stress. In addition, hermetic packages have fine/gross leak readouts. This test is to detect mechanical reliability problems and microcracks.





NOTE:
Output toggle frequency is device maximum frequency.

Figure 1. EPLD Lifetest/Burn-In Bias and Timing Diagram (85C220 Used as Example)

85/85 Humidity—High temperature/humidity testing is performed to evaluate moisture resistance characteristics of plastic-encapsulated components. A 2000-hour test is performed under static bias conditions at 85°C/85 percent relative humidity with nominal voltages. To maximum metal corrosion conditions, the biasing configuration is either under low power or no power, with alternate pins biased at +5V or 0V.

Steam—This test consists of exposing parts to water vapor at 121°C and at 2 atmospheres. The devices are functionally tested before and after the test. This test is used to highly accelerate failure mechanisms effecting the bonds, bond pads, and passivation.

ESD Testing—This test is performed to validate the products tolerance to Electro Static Discharge damage. All products incorporate ESD protection networks where needed to ensure the Intel corporate goals of military and charged device ESD testing are met. The military test uses the MIL STD 883 test criteria, while the charged device testing is performed to further validate protection occurring during mechanical handling.

Programmability—Device programmability is routinely monitored through the process of programming the devices for product monitors and qualifications Programmability is a distinct part of a product qualification. All voltage combinations are verified. Program margin is measured and tested on ≥90% of Intel EPLD EPROM cells.

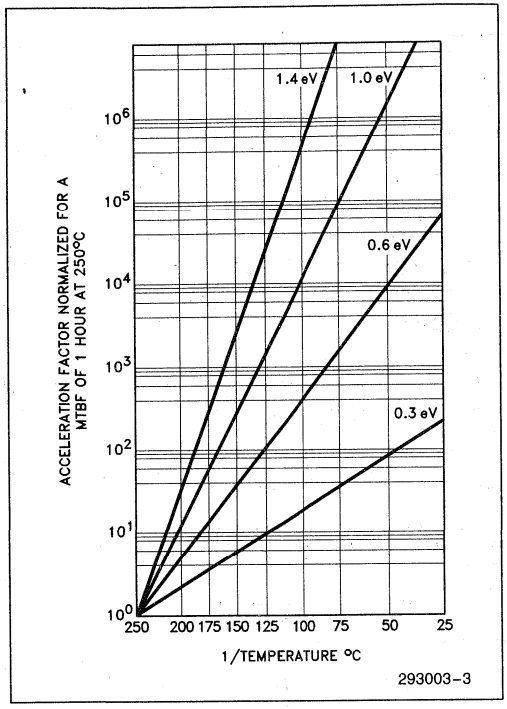


Figure 2. Arrhenius Plot

REFERENCES

1. S. Rosenberg, D. Crook, B. Euzent, "16th Annual Proceedings of the International Reliability Physics Symposium," pp 19–25, 1978.
2. J. Caywood, B. Euzent, B. Shiner, "Data Retention in EPROMs, " 1980 IEEE International Reliability Physics Symposium.
3. S. Rosenberg, B. Euzent, "HMOS Reliability" Reliability Report RR-18, Intel Corporation, 1979.
4. N. Mielke, "New EPROM Data-Loss Mechanisms," 1983. International Reliability Physics Symposium.
5. R. M. Alexander, "Calculating Failure Rates From Stress Data," April 1984 International Reliability Physics Symposium.
6. "EPROM Reliability Data Summary", Reliability Report RR-35, Intel Corporation, 1989, (updated regularly).
7. "EPROM Reliability Data Summary CHMOS IIIE", Reliability Report RR-67, Intel Corporation, 1990, order number: 293004. (Updated Regularly).
8. "Intel Components Quality/Reliability Handbook", 1989, order number: 210997-004.
9. D. Baglee, L. Nanneman, C. Huang, "28th Annual Proceedings of the International Reliability Physics Symposium", pp. 12–18, 1990.
10. E. Anolick, G. Nelson, "17th Annual Proceedings of the International Reliability Physics Symposium", 1979.

Reliability Data Summary

The following data is an accumulation of recent qualification and monitor program results. Failure rate calculation methods listed in Appendix A were used to arrive at the tabularized failure rates. CERDIP lifetest stresses are used to calculate the failure rate for each product and should also be used to indicate the failure rate of plastic products.

In reviewing the reliability data as presented, questions may arise as to why lot sizes often decrease from one test to another without a corresponding number of identified failures. This is due to a variety of factors. Many tests require smaller sample sizes and as a result all parts from a previous test do not necessarily flow through to a succeeding test.

In addition, various parts are pulled from a sample lot when mechanical or handler problems cause failures to occur. These "failures" are not a result of the specific test just completed. They are removed from the sample lot size and are not included in any failure rate calculation. It can also happen that a particular test is done incorrectly through human error or faulty test equipment and these suspected "invalid" failures are put aside for retesting at a later date, decreasing the lot size for a succeeding test. If these parts are found to be truly defective, they are treated as failures and listed. If they test out properly, they are removed from any calculation data base.

5C032

Device: 5C032
Organization 8 Macrocells
Pinout: 20 Lead, 300 mil CERDIP and PDIP
Die Size: 86 x 93 mils
Process: CHMOS IIE, P424
Programming Voltage: 12.5V
Technology: CMOS

Table 1. Reliability Data

Package	Burn-In 48 Hours	125°C Dynamic Lifetest		
		168 Hours	500 Hours	1K Hours
CERDIP	0/2956	0/2956	0/299	2/283
PDIP	1/2661	0/2660	0/149	0/146
TOTALS	1/5617 A	0/5616	0/448	2/429 B
Additional Readouts: CERDIP	0/281 @ 2K Hours Lifetest			

Package	125°C High Voltage Dynamic Lifetest (6.5V)		
	48 Hours	168 Hours	500 Hours
CERDIP	0/200	0/200	0/179
PDIP	0/150	1/150	0/149
TOTALS	0/350	1/350 C	0/328
Additional Readouts: CERDIP	0/129 @ 1K Hours High Voltage Lifetest		
PDIP	0/146 @ 1K Hours High Voltage Lifetest		

Package	Data Retention Bake (CERDIP @ 250°C, PDIP @ 140°C)		
	48 Hours	168 Hours	500 Hours
CERDIP	2/277	0/146	0/146
PDIP	0/300	1/298	0/298
TOTALS	2/577 D	0/444	0/444
Additional Readouts: PDIP	0/298 @ 1K Hours, 140°C Bake		

3

Table 2. Additional Qualification Stress Information

Package	Temperature Cycling (Condition "C"—150°C to −65°C)		
	200 Cycles	500 Cycles	1K Cycles
CERDIP PDIP	0/232 0/150	0/75 0/149	0/75 0/149
TOTALS	0/382	0/224	0/224

Package	Thermal Shock (Condition "C"—150°C to −65°C)		
	50 Cycles	200 Cycles	500 Cycles
PDIP	0/150	0/150	0/150
TOTALS	0/150	0/150	0/150

Package	85°C/85% Relative Humidity		
	168 Hours	500 Hours	1K Hours
PDIP	0/263	0/263	1/263
TOTALS	0/263	0/263	1/263 E

Package	Steam (121°C, 2 ATM)	
	96 Hours	168 Hours
PDIP	0/262	0/262
Additional Readouts: PDIP	0/262 @ 500 Hours of Steam	

Table 3. Failure Rate Predictions (CERDIP)

125°C Actual Device Hours	Ea (eV)	Equivalent Hours 55°C	Equivalent Hours 70°C	# Fail	Fail Rate %/1K Hours (60% UCL) 55°C	Fail Rate %/1K Hours (60% UCL) 70°C
7.36×10^5	0.3 B.I.	4.00×10^6	2.60×10^6	0		
1.58×10^5	0.3 HVLT * VAF	4.72×10^7	3.07×10^7	0		
			Total 0.3 eV Failures =	0	0.0018	0.0028
7.36×10^5	0.5 B.I.	1.23×10^7	6.03×10^6	0		
7.99×10^4	0.5 Bake	5.81×10^7	2.68×10^7	1		
1.58×10^5	0.5 HVLT	2.64×10^6	1.29×10^6	0		
			Total 0.5 eV Failures =	1	0.0028	0.0059
7.36×10^5	0.6 B.I.	2.17×10^7	9.18×10^6	0		
7.99×10^4	0.6 Bake	2.17×10^8	8.59×10^7	1		
1.58×10^5	0.6 HVLT	4.64×10^6	1.97×10^6	0		
			Total 0.6 eV Failures =	1	0.0008	0.0039
7.36×10^5	1.0 B.I.	2.07×10^8	4.94×10^7	2		
1.58×10^5	1.0 HVLT	4.43×10^7	1.06×10^7	0		
			Total 1.0 eV Failures =	2	0.0012	0.0052
			Combined Failure Rate:		0.0066	0.0159
			FITs:		66	159
48 Hour BI Infant Mortality = See end of Table 4						

Other Data (CERDIP)

Theta Ja =	83°C/W	**Temp with Theta Ja Degree Kelvin**	
V_{CC} =	5.25V	T(55°C) =	341.07 K
I_{CC} @ 55°C =	30 mA	T(70°C =	356.07 K
I_{CC} @ 70°C =	30 mA	T(125°C) =	408.89 K
I_{CC} @ 125°C =	25 mA	T(140°C) =	413 K

Boltzman's Constant = K = 8.62×10^{-5} eV/K

Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLT) Activation Energy	55°C	70°C
0.3	5.43	3.53
0.5	16.76	8.19
0.6	29.46	12.48
1.0	280.99	67.11

Bake (140°C) Activation Energy	55°C	70°C
0.3	n/a	n/a
0.5	727.61	335.91
0.6	2718.18	1075.15
1.0	n/a	n/a

Table 4. Failure Rate Predictions (PDIP)

125°C Actual Device Hours	Ea (eV)	Equivalent Hours 55°C	Equivalent Hours 70°C	# Fail	Fail Rate %/1K Hours (60% UCL) 55°C	Fail Rate %/1K Hours (60% UCL) 70°C
4.42×10^5 1.48×10^5	0.3 B.I. 0.3 HVLT * VAF	2.28×10^6 4.20×10^7	1.50×10^6 2.76×10^7	0 0		
		Total 0.3 eV Failures =		0	0.0021	0.0032
4.42×10^5 1.48×10^5 1.48×10^5	0.5 B.I. 0.5 Bake 0.5 HVLT	6.81×10^6 1.14×10^7 2.28×10^6	3.38×10^6 5.25×10^6 1.13×10^6	0 1 0		
		Total 0.5 eV Failures =		1	0.0099	0.0207
4.42×10^5 1.48×10^5 1.48×10^5	0.6 B.I. 0.6 Bake 0.6 HVLT	1.18×10^7 2.35×10^7 3.93×10^6	5.08×10^6 9.31×10^6 1.70×10^6	0 0 0		
		Total 0.6 eV Failures =		0	0.0	0.0
4.42×10^5 1.48×10^5	1.0 B.I. 1.0 HVLT	1.05×10^8 3.51×10^7	2.59×10^7 8.65×10^6	0 0		
		Total 1.0 eV Failures =		0	0.0	0.0
			Combined Failure Rate: FITs:		0.0120 120	0.0239 239
48 Hour BI Infant Mortality (CERDIP and PDIP) = 1/5617 = 0.0178% = 178 DPM						

Other Data (PDIP)

		Temp with Theta Ja Degree Kelvin	
Theta Ja =	109°C/W		
V_{CC} =	5.25V	T(55°C) =	345.17 K
I_{CC} @ 55°C =	30 mA	T(70°C) =	360.17 K
I_{CC} @ 70°C =	30 mA	T(125°C) =	412.31 K
I_{CC} @ 125°C =	25 mA	T(140°C) =	413 K

Boltzman's Constant = K = 8.62×10^{-5} eV/K

Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLT)			Bake (140°C)		
Activation Energy	**55°C**	**70°C**	**Activation Energy**	**55°C**	**70°C**
0.3	5.16	3.39	0.3	n/a	n/a
0.5	15.41	7.66	0.5	38.00	17.54
0.6	26.62	11.50	0.6	78.65	31.11
1.0	237.39	58.60	1.0	n/a	n/a

Voltage Acceleration Factor (VAF) for HVLT on this process is 55.

NOTE: FIT = Failure in Time. 1 FIT = 1 failure per 1×10^9 device hours.

Failure Analysis

A. 1 leakage – 0.5 eV
B. 2 leakage – 1.0 eV
C. 1 leakage – 0.5 eV
D. 1 SBCL – 0.6 eV
   1 passivation defect – 0.5 eV
E. 1 passivation defect – 0.5 eV

5C060

Device: 5C060
Organization 16 Macrocells
Pinout: 24 Lead, 300 mil CERDIP and PDIP (28 Lead PLCC)
Die Size: 135 x 141 mils
Process: CHMOS IIE, P424
Programming Voltage: 12.5V
Technology: CMOS

Table 1. Reliability Data

Package	Burn-In 48 Hours	125°C Dynamic Lifetest		
		168 Hours	500 Hours	1K Hours
CERDIP				
Fab 1	0/3969	0/3969	0/371	0/354
Fab 4	1/2500	3/2499	0/100	0/99
PDIP				
Fab 1		0/351	0/351	0/351
TOTALS	1/6469 A	3/6819 B	0/882	0/804
Additional Readouts:				
CERDIP				
Fab 4	0/99 @ 2K Hours Lifetest			
PDIP				
Fab 1	0/351 @ 2K Hours Lifetest			

Package	125°C High Voltage Dynamic Lifetest (6.5V)		
	48 Hours	168 Hours	500 Hours
CERDIP			
Fab 1	0/224	0/201	2/198
Fab 4	0/100	1/100	0/100
TOTALS	0/324	1/301	2/298 C

Package	Data Retention Bake (CERDIP @ 250°C)		
	48 Hours	168 Hours	500 Hours
CERDIP			
Fab 1	0/203	0/202	0/202
Fab 4	0/50	0/50	0/50
TOTALS	0/253	0/252	0/252
Additional Readouts:			
CERDIP			
Fab 1	0/127 @ 1K Hours, 250°C Bake		

Table 2. Additional Qualification Stress Information

Package	Temperature Cycling (Condition "C"—150°C to −65°C)		
	200 Cycles	500 Cycles	1K Cycles
CERDIP PDIP	0/307 0/350	1/307 0/350	0/302 —
TOTALS	0/657	1/657 D	0/302

Package	Thermal Shock (Condition "C"—150°C to −65°C)		
	50 Cycles	200 Cycles	500 Cycles
CERDIP	0/155	0/155	1/155
TOTALS	0/155	0/155	1/155 E
Additional Readouts: CERDIP 0/154 @ 1K Cycles			

Package	Steam (121°C, 2 ATM)	
	96 Hours	168 Hours
PDIP	0/100	0/100
PLCC	0/150	0/150
TOTALS	0/250	0/250
Additional Readouts: PDIP 0/100 @ 336 Hours of Steam		

Table 3. Failure Rate Predictions (5C060-CERDIP-Fab 1)

125°C Actual Device Hours	Ea (eV)	Equivalent Hours		# Fail	Fail Rate %/1K Hours (60% UCL)	
		55°C	70°C		55°C	70°C
7.76×10^5 1.10×10^5	0.3 B.I. 0.3 HVLT * VAF	3.93×10^6 2.80×10^7	2.62×10^6 1.87×10^7	0 2		
			Total 0.3 eV Failures =	2	0.0097	0.0146
7.76×10^5 1.65×10^5 1.10×10^5	0.5 B.I. 0.5 Bake 0.5 HVLT	1.16×10^7 1.20×10^8 1.50×10^6	5.89×10^6 5.55×10^7 7.63×10^5	0 0 0		
			Total 0.5 eV Failures =	0	0.0000	0.0000
7.76×10^5 1.65×10^5 1.10×10^5	0.6 B.I. 0.6 Bake 0.6 HVLT	1.99×10^7 4.49×10^8 2.58×10^6	8.83×10^6 1.78×10^8 1.14×10^6	0 0 0		
			Total 0.6 eV Failures =	0	0.0000	0.0000
7.76×10^5 1.10×10^5	1.0 B.I. 1.0 HVLT	1.73×10^8 2.25×10^7	4.47×10^7 5.79×10^6	0 0		
			Total 1.0 eV Failures =	0	0.0000	0.0000
			Combined Failure Rate: FITs:		0.0097 97	0.0146 146
				48 Hour BI Infant Mortality = See end of Table 4		

Table 4. Failure Rate Predictions (5C060-CERDIP-Fab 4)

125°C Actual Device Hours	Ea (eV)	Equivalent Hours		# Fail	Fail Rate %/1K Hours (60% UCL)	
		55°C	70°C		55°C	70°C
4.81×10^5	0.3 B.I.	2.44×10^6	1.62×10^6	2		
5.00×10^4	0.3 HVLT * VAF	1.39×10^7	9.27×10^6	0		
			Total 0.3 eV Failures =	2	0.0190	0.0285
4.81×10^5	0.5 B.I.	7.20×10^6	3.65×10^6	1		
2.52×10^4	0.5 Bake	1.83×10^7	8.46×10^6	0		
5.00×10^4	0.5 HVLT	7.47×10^5	3.79×10^5	0		
			Total 0.5 eV Failures =	1	0.0077	0.0162
4.81×10^5	0.6 B.I.	1.24×10^7	5.48×10^6	0		
2.52×10^4	0.6 Bake	6.85×10^7	2.71×10^7	0		
5.00×10^4	0.6 HVLT	1.28×10^6	5.69×10^5	0		
			Total 0.6 eV Failures =	0	0.0	0.0
4.81×10^5	1.0 B.I.	1.08×10^8	2.77×10^7	0		
5.00×10^5	1.0 HVLT	1.12×10^7	2.88×10^6	0		
			Total 1.0 eV Failures =	0	0.0	0.0
			Combined Failure Rate:		0.0267	0.0447
			FITs:		267	447
48 Hour BI Infant Mortality (D5C060-F1 and D5C060-F4) = 1/6469 = 0.0155% = 155 DPM						

Other Data (CERDIP)

		Temp with Theta Ja Degree Kelvin	
Theta Ja =	54°C/W		
V_{CC} =	5.25V	T(55°C) =	350.68 K
I_{CC} @ 55°C =	80 mA	T(70°C =	365.68 K
I_{CC} @ 70°C =	80 mA	T(125°C) =	419.26 K
I_{CC} @ 125°C =	75 mA	T(140°C) =	523 K

Boltzman's Constant = K = 8.62×10^{-5} eV/K

Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLT)			Bake (250°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	5.07	3.37	0.3	n/a	n/a
0.5	14.94	7.58	0.5	727.61	335.91
0.6	25.66	11.37	0.6	2718.18	1075.15
1.0	223.24	57.52	1.0	n/a	n/a

Other Data (PDIP)

		Temp with Theta Ja Degree Kelvin	
Theta Ja =	67°C/W		
V_{CC} =	5.25V	T(55°C) =	356.14 K
I_{CC} @ 55°C =	80 mA	T(70°C =	371.14 K
I_{CC} @ 70°C =	80 mA	T(125°C) =	424.38 K
I_{CC} @ 125°C =	75 mA	T(140°C) =	523 K

Boltzman's Constant = K = 8.62×10^{-5} eV/K

Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLT)			Bake (250°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	4.81	3.24	0.3	n/a	n/a
0.5	13.70	7.10	0.5	38.00	17.54
0.6	23.13	10.50	0.6	78.65	31.11
1.0	187.72	50.36	1.0	n/a	n/a

Other Data (PLCC)

		Temp with Theta Ja Degree Kelvin	
Theta Ja =	61°C/W		
V_{CC} =	5.25V	T(55°C) =	353.62 K
I_{CC} @ 55°C =	80 mA	T(70°C) =	368.62 K
I_{CC} @ 70°C =	80 mA	T(125°C) =	422.02 K
I_{CC} @ 125°C =	75 mA	T(140°C) =	413 K

Boltzman's Constant = K = 8.62×10^{-5} eV/K

Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLT)			Bake (140°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	4.92	3.30	0.3	n/a	n/a
0.5	14.25	7.31	0.5	38.00	17.54
0.6	24.25	10.89	0.6	78.65	31.11
1.0	203.16	53.51	1.0	n/a	n/a

Voltage Acceleration Factor (VAF) for HVLT on this process is 55.

NOTE:
FIT = Failure in Time. 1 FIT = 1 failure per 1×10^9 device hours.

Failure Analysis

A. 1 oxide defect − 0.3 eV

B. 2 oxide defects − 0.3 eV
1 metal defect − 0.5 eV

C. 2 oxide defects − 0.3 eV

D. 1 passivation defect − 0.5 eV

E. 1 passivation defect − 0.5 eV

5C090

Device: 5C090
Organization 24 Macrocells
Pinout: 40 Lead, 600 mil CERDIP and PDIP (44 Lead PLCC)
Die Size: 166 x 181 mils
Process: CHMOS IIE, P424
Programming Voltage: 12.5V
Technology: CMOS

Table 1. Reliability Data

Package	Burn-In 48 Hours	125°C Dynamic Lifetest		
		168 Hours	500 Hours	1K Hours
CERDIP				
Fab 1	1/2123	2/2085	0/170	0/170
Fab 4	3/3934	3/3931	0/190	0/190
TOTALS	4/6057 A	2/6016 B	0/360	0/360
Additional Readouts:				
CERDIP				
Fab 1	0/170 @ 2K Hours Lifetest			
Fab 4	0/95 @ 2K Hours Lifetest			

Package	125°C High Voltage Dynamic Lifetest (6.5V)		
	48 Hours	168 Hours	500 Hours
CERDIP			
Fab 1	0/170	0/170	0/156
Fab 4	0/195	0/195	0/195
TOTALS	0/365	0/365 C	0/351

Package	Data Retention Bake (CERDIP @ 250°C)		
	48 Hours	168 Hours	500 Hours
CERDIP			
Fab 1	0/66	0/66	0/66
Fab 4	0/99	0/99	0/99
TOTALS	0/165	0/165	0/165

Table 2. Additional Qualification Stress Information

Package	Temperature Cycling (Condition "C"—150°C to −65°C)		
	200 Cycles	500 Cycles	1K Cycles
CERDIP	0/149	0/141	1/67 D

Package	Thermal Shock (Condition "C"—150°C to −65°C)		
	50 Cycles	200 Cycles	500 Cycles
CERDIP	0/50	0/50	1/50 E

Package	85°C/85% Relative Humidity		
	168 Hours	500 Hours	1K Hours
PLCC Fab 4	0/172	0/172	0/172
Additional Readouts PLCC Fab 4	0/172 @ 2K Hours		

Package	Steam (121°C, 2 ATM)	
	96 Hours	168 Hours
PDIP	0/153	0/153
PLCC	3/474 F	0/471
TOTALS	3/627	0/624
Additional Readouts: PDIP	0/300 @ 500 Hours of Steam	

Table 3. Failure Rate Predictions (5C090-CERDIP-Fab 1)

125°C Actual Device Hours	Ea (eV)	Equivalent Hours		# Fail	Fail Rate %/1K Hours (60% UCL)	
		55°C	70°C		55°C	70°C
5.61×10^5 8.04×10^4	0.3 B.I. 0.3 HVLT * VAF	2.96×10^6 2.33×10^7	1.94×10^6 1.53×10^7	0 1		
		Total 0.3 eV Failures =		1	0.0077	0.0117
5.61×10^5 3.33×10^4 8.04×10^4	0.5 B.I. 0.5 Bake 0.5 HVLT	8.96×10^6 2.42×10^7 1.28×10^6	4.45×10^6 1.12×10^7 6.37×10^5	0 0 0		
		Total 0.5 eV Failures =		0	0.0000	0.0000
5.61×10^5 3.33×10^4 8.04×10^4	0.6 B.I. 0.6 Bake 0.6 HVLT	1.56×10^7 9.04×10^7 2.23×10^6	6.73×10^6 3.58×10^7 9.63×10^5	2 0 0		
		Total 0.6 eV Failures =		2	0.0029	0.0071
5.61×10^5 8.04×10^4	1.0 B.I. 1.0 HVLT	1.43×10^8 2.05×10^7	3.53×10^7 5.05×10^6	0 0		
		Total 1.0 eV Failures =		0	0.0000	0.0000
			Combined Failure Rate: FITs:		0.0106 106	0.0188 188
			48 Hour BI Infant Mortality = See end of Table 4			

Table 4. Failure Rate Predictions (5C090-CERDIP-Fab 4)

125°C Actual Device Hours	Ea (eV)	Equivalent Hours		# Fail	Fail Rate %/1K Hours (60% UCL)	
		55°C	70°C		55°C	70°C
7.25×10^5 9.75×10^4	0.3 B.I. 0.3 HVLT * VAF	3.82×10^6 2.83×10^7	2.51×10^6 1.86×10^7	0 0		
		Total 0.3 eV Failures =		0	0.0029	0.0043
7.25×10^5 4.99×10^4 9.75×10^4	0.5 B.I. 0.5 Bake 0.5 HVLT	1.16×10^7 3.63×10^7 1.56×10^6	5.74×10^6 1.68×10^7 7.73×10^5	0 0 0		
		Total 0.5 eV Failures =		0	0.0000	0.0000
7.25×10^5 4.99×10^4 9.75×10^4	0.6 B.I. 0.6 Bake 0.6 HVLT	2.01×10^7 1.36×10^8 2.71×10^6	8.69×10^6 5.36×10^7 1.17×10^6	0 0 0		
		Total 0.6 eV Failures =		0	0.0	0.0
7.25×10^5 9.75×10^4	1.0 B.I. 1.0 HVLT	1.85×10^8 2.48×10^7	4.55×10^7 6.12×10^6	0 0		
		Total 1.0 eV Failures =		0	0.0	0.0
			Combined Failure Rate:		0.0029	0.0043
			FITs:		29	43
48 Hour BI Infant Mortality (D5C090-F1 and D5C090-F4) = 4/6057 = 0.0660% = 660 DPM						

Other Data (CERDIP)

		Temp with Theta Ja Degree Kelvin	
Theta Ja =	36°C/W		
V_{CC} =	5.25V	T(55°C) =	345.01 K
I_{CC} @ 55°C =	90 mA	T(70°C) =	360.01 K
I_{CC} @ 70°C =	90 mA	T(125°C) =	413.12 K
I_{CC} @ 125°C =	80 mA	T(140°C) =	523 K

Boltzman's Constant = K = 8.62×10^{-5} eV/K

Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLT)			Bake (250°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	5.27	3.46	0.3	n/a	n/a
0.5	15.96	7.93	0.5	727.61	335.91
0.6	27.78	11.99	0.6	2718.18	1075.15
1.0	254.79	62.82	1.0	n/a	n/a

3

Other Data (PDIP)

		Temp with Theta Ja Degree Kelvin	
Theta Ja =	48°C/W		
V_{CC} =	5.25V	T(55°C) =	350.68 K
I_{CC} @ 55°C =	90 mA	T(70°C) =	365.68 K
I_{CC} @ 70°C =	90 mA	T(125°C) =	418.16 K
I_{CC} @ 125°C =	80 mA	T(140°C) =	413 K

Boltzman's Constant = K = 8.62×10^{-5} eV/K

Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLT)			Bake (140°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	4.96	3.30	0.3	n/a	n/a
0.5	14.41	7.31	0.5	38.00	17.54
0.6	24.56	10.89	0.6	78.65	31.11
1.0	207.54	53.47	1.0	n/a	n/a

Other Data (PLCC)

		Temp with Theta Ja Degree Kelvin	
Theta Ja =	48°C/W		
V_{CC} =	5.25V	T(55°C) =	350.68 K
I_{CC} @ 55°C =	90 mA	T(70°C) =	365.68 K
I_{CC} @ 70°C =	90 mA	T(125°C) =	418.16 K
I_{CC} @ 125°C =	80 mA	T(140°C) =	413 K

Boltzman's Constant = K = 8.62×10^{-5} eV/K

Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLT)			Bake (140°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	4.96	3.30	0.3	n/a	n/a
0.5	14.41	7.31	0.5	38.00	17.54
0.6	24.56	10.89	0.6	78.65	31.11
1.0	207.54	53.47	1.0	n/a	n/a

Voltage Acceleration Factor (VAF) for HVLT on this process is 55.

NOTE:
FIT = Failure in Time. 1 FIT = 1 failure per 1×10^9 device hours.

Failure Analysis

A. 1 SBCL – 0.6 eV
   1 polysilicon defect – 0.5 eV
   1 metal defect – 0.5 eV
   1 oxide defect – 0.3 eV
B. 2 MBCL – 0.6 eV
C. 1 oxide defect – 0.3 eV
D. 1 open metal – 0.5 eV
E. 1 hermeticity
F. 3 passivation defects – 0.5 eV

5C180

Device: 5C180
Organization 48 Macrocells
Pinout: 68 Lead PLCC, CERQUAD, and PGA
Die Size: 265 x 182 mils
Process: CHMOS IIE, P424
Programming Voltage: 12.5V
Technology: CMOS

Table 1. Reliability Data

Package	Burn-In 48 Hours	125°C Dynamic Lifetest		
		168 Hours	500 Hours	1K Hours
CERQUAD	0/378	0/376	0/102	0/102
TOTALS	0/378	0/376	0/102	0/102

Package	125°C High Voltage Dynamic Lifetest (6.5V)		
	48 Hours	168 Hours	500 Hours
CERQUAD	0/50	0/43	0/39
TOTALS	0/50	0/43	0/39

Package	Data Retention Bake (CERDIP @ 250°C)		
	48 Hours	168 Hours	500 Hours
CERQUAD	0/107	0/100	0/100
TOTALS	0/107	0/100	0/100

Table 2. Additional Qualification Stress Information

Package	Temperature Cycling (Condition "C"—150°C to −65°C)		
	200 Cycles	500 Cycles	1K Cycles
CERQUAD	0/157	0/157	1/146
PGA	0/128	0/128	0/128
PLCC	0/123	0/123	0/123
TOTALS	0/408	0/408	1/397 A

Package	Thermal Shock (Condition "C"—150°C to −65°C)		
	50 Cycles	200 Cycles	500 Cycles
PGA	0/117	0/117	0/177
TOTALS	0/117	0/117	0/177

Package	Steam (121°C, 2 ATM)	
	96 Hours	168 Hours
PLCC	0/77	0/77
TOTALS	0/77	0/77

Additional Readouts:
PLCC 0/77 @ 500 Hours of Steam

3

Table 3. Failure Rate Predictions (5C180-CERQUAD)

125°C Actual Device Hours	Ea (eV)	Equivalent Hours		# Fail	Fail Rate %/1K Hours (60% UCL)	
		55°C	70°C		55°C	70°C
1.30×10^5 2.05×10^4	0.3 B.I. 0.3 HVLT * VAF	5.81×10^5 5.04×10^6	3.94×10^5 3.42×10^6	0 0		
		Total 0.3 eV Failures =		0	0.0163	0.0240
1.30×10^5 5.07×10^4 2.05×10^4	0.5 B.I. 0.5 Bake 0.5 HVLT	1.58×10^6 3.69×10^7 2.49×10^5	8.25×10^5 1.70×10^7 1.30×10^5	0 0 0		
		Total 0.5 eV Failures =		0	0.0000	0.0000
1.30×10^5 5.07×10^4 2.05×10^4	0.6 B.I. 0.6 Bake 0.6 HVLT	2.60×10^6 1.38×10^8 4.10×10^5	1.19×10^6 5.45×10^7 1.88×10^5	0 0 0		
		Total 0.6 eV Failures =		0	0.0	0.0
1.30×10^5 2.05×10^4	1.0 B.I. 1.0 HVLT	1.91×10^7 3.02×10^6	5.24×10^6 8.26×10^5	0 0		
		Total 1.0 eV Failures =		0	0.0	0.0
			Combined Failure Rate: FITs:		0.0163 163	0.0240 240
48 Hour BI Infant Mortality = 0/378 = 0.00% = 0 DPM						

Other Data (CERQUAD)

		Temp with Theta Ja Degree Kelvin	
Theta Ja =	42°C/W		
V_{CC} =	5.25V	T(55°C) =	358.87 K
I_{CC} @ 55°C =	140 mA	T(70°C) =	373.87 K
I_{CC} @ 70°C =	140 mA	T(125°C) =	424.46 K
I_{CC} @ 125°C =	120 mA	T(140°C) =	523 K

Boltzman's Constant = K = 8.62×10^{-5} eV/K

Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLT)			Bake (250°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	4.47	3.03	0.3	n/a	n/a
0.5	12.14	6.35	0.5	727.61	335.91
0.6	19.99	9.19	0.6	2718.18	1075.15
1.0	147.28	40.29	1.0	n/a	n/a

Other Data (PLCC)

		Temp with Theta Ja Degree Kelvin	
Theta Ja =	38°C/W		
V_{CC} =	5.25V	T(55°C) =	355.93 K
I_{CC} @ 55°C =	140 mA	T(70°C) =	370.93 K
I_{CC} @ 70°C =	140 mA	T(125°C) =	421.94 K
I_{CC} @ 125°C =	120 mA	T(140°C) =	413 K

Boltzman's Constant = K = 8.62×10^{-5} eV/K

Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLT)			Bake (140°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	4.61	3.11	0.3	n/a	n/a
0.5	12.78	6.62	0.5	38.00	17.54
0.6	21.28	9.65	0.6	78.65	31.11
1.0	163.38	43.76	1.0	n/a	n/a

3

Other Data (PGA)

		Temp with Theta Ja Degree Kelvin	
Theta Ja =	28.5°C/W		
V_{CC} =	5.25V	T(55°C) =	348.95 K
I_{CC} @ 55°C =	140 mA	T(70°C) =	363.95 K
I_{CC} @ 70°C =	140 mA	T(125°C) =	415.96 K
I_{CC} @ 125°C =	120 mA	T(140°C) =	523 K

Boltzman's Constant = K = 8.62×10^{-5} eV/K

Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLT)			Bake (140°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	4.98	3.30	0.3	n/a	n/a
0.5	14.53	7.33	0.5	727.61	335.91
0.6	24.82	10.91	0.6	2718.18	1075.15
1.0	211.14	53.68	1.0	n/a	n/a

Voltage Acceleration Factor (VAF) for HVLT on this process is 55.

NOTE:
FIT = Failure in Time. 1 FIT = 1 failure per 1×10^9 device hours.

Failure Analysis

A. 1 hermiticity

5AC312

Device: 5AC312
Organization 12 Macrocells
Pinout: 24 Lead, 300 mil CERDIP and PDIP (28 Lead PLCC)
Die Size: 130 x 189 mils
Process: CHMOS IIIE
Programming Voltage: 12.5V, P429.3
Technology: CMOS

Table 1. Reliability Data

Package	Burn-In 48 Hours	125°C Dynamic Lifetest		
		168 Hours	500 Hours	1K Hours
CERDIP	1/3960	0/2955	0/183	0/183
TOTALS	1/3960 A	0/2955	0/183	0/183
Additional Readouts: CERDIP	0/183 @ 2K Hours Lifetest			

Package	125°C High Voltage Dynamic Lifetest (7.0V)		
	48 Hours	168 Hours	500 Hours
CERDIP	0/97	0/93	0/92
TOTALS	0/97	0/93	0/92
Additional Readouts: CERDIP	0/80 @ 1K Hours High Voltage Lifetest		

Package	Data Retention Bake (CERDIP @ 250°C)		
	48 Hours	168 Hours	500 Hours
CERDIP	0/100	0/99	0/99
TOTALS	0/100	0/99	0/99
Additional Readouts: CERDIP	0/96 @ 1K Hours, 250°C Bake		

Table 2. Additional Qualification Stress Information

Package	Temperature Cycling (Condition "C"—150°C to −65°C)		
	200 Cycles	500 Cycles	1K Cycles
PDIP	0/234	0/234	0/234
TOTALS	0/234	0/234	0/234

Package	85°C/85% Relative Humidity		
	168 Hours	500 Hours	1K Hours
PDIP	0/219	0/219	0/219
TOTALS	0/219	0/219	0/219
Additional Readouts: PDIP	0/219 @ 2K Hours		

Package	Thermal Shock (Condition "C"—150°C to −65°C)		
	50 Cycles	200 Cycles	500 Cycles
PDIP	0/231	0/231	0/231
TOTALS	0/231	0/231	0/231

Package	Steam (121°C, 2 ATM)	
	96 Hours	168 Hours
PDIP	0/393	0/393
PLCC	0/304	0/304
TOTALS	0/697	0/697
Additional Readouts: PDIP	0/234 @ 336 Hours of Steam	

3

Table 3. Failure Rate Predictions (5AC312-CERDIP-Fab 1)

125°C Actual Device Hours	Ea (eV)	Equivalent Hours		# Fail	Fail Rate %/1K Hours (60% UCL)	
		55°C	70°C		55°C	70°C
6.90×10^5 8.64×10^5	0.3 B.I. 0.3 HVLT * VAF	3.42×10^6 4.33×10^7	2.26×10^6 2.86×10^7	0 0		
			Total 0.3 eV Failures =	0	0.0155	0.0235
6.90×10^5 9.79×10^4 8.64×10^4	0.5 B.I. 0.5 Bake 0.5 HVLT	9.96×10^6 7.13×10^7 1.25×10^6	5.00×10^6 3.29×10^7 6.26×10^5	0 0 0		
			Total 0.5 eV Failures =	0	0.0000	0.0000
6.90×10^5 9.79×10^4 8.64×10^4	0.6 B.I. 0.6 Bake 0.6 HVLT	1.70×10^7 2.66×10^8 2.13×10^6	7.42×10^6 1.05×10^8 9.29×10^5	0 0 0		
			Total 0.6 eV Failures =	0	0.0000	0.0000
6.90×10^5 8.64×10^4	1.0 B.I. 1.0 HVLT	1.44×10^8 1.80×10^7	3.62×10^7 4.53×10^6	0 0		
			Total 1.0 eV Failures =	0	0.0000	0.0000
				Combined Failure Rate: FITs:	0.0021 21	0.0032 32
48 Hour BI Infant Mortality = 1/3960 = 0.0253% = 253 DPM						

Other Data (CERDIP)

		Temp with Theta Ja Degree Kelvin	
Theta Ja =	47°C/W		
V_{CC} =	5.25V	T(55°C) =	348 K
I_{CC} @ 55°C =	80 mA	T(70°C) =	363 K
I_{CC} @ 70°C =	80 mA	T(125°C) =	414 K
I_{CC} @ 125°C =	65 mA	T(250°C) =	523 K

Boltzman's Constant = K = 8.62×10^{-5} eV/K

Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLT)			Bake (250°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	4.96	3.28	0.3	n/a	n/a
0.5	14.43	7.24	0.5	727.61	335.91
0.6	24.61	10.76	0.6	2718.18	1075.15
1.0	208.28	52.47	1.0	n/a	n/a

Other Data (PDIP)

		Temp with Theta Ja Degree Kelvin	
Theta Ja =	52°C/W		
V_{CC} =	5.25V	T(55°C) =	349.84 K
I_{CC} @ 55°C =	80 mA	T(70°C) =	364.84 K
I_{CC} @ 70°C =	80 mA	T(125°C) =	415.75 K
I_{CC} @ 125°C =	65 mA	T(140°C) =	413 K

Boltzman's Constant = K = 8.62×10^{-5} eV/K

Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLT)			Bake (140°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	4.84	3.21	0.3	n/a	n/a
0.5	13.83	7.00	0.5	38.00	17.54
0.6	23.39	10.33	0.6	78.65	31.11
1.0	191.27	48.96	1.0	n/a	n/a

Other Data (PLCC)

		Temp with Theta Ja Degree Kelvin	
Theta Ja =	56°C/W		
V_{CC} =	5.25V	T(55°C) =	351.52 K
I_{CC} @ 55°C =	80 mA	T(70°C) =	366.52 K
I_{CC} @ 70°C =	80 mA	T(125°C) =	417.11 K
I_{CC} @ 125°C =	65 mA	T(140°C) =	413 K

Boltzman's Constant = K = 8.62×10^{-5} eV/K

Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLT)			Bake (140°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	4.74	3.16	0.3	n/a	n/a
0.5	13.37	6.81	0.5	38.00	17.54
0.6	22.46	9.99	0.6	78.65	31.11
1.0	178.85	46.37	1.0	n/a	n/a

Voltage Acceleration Factor (VAF) for HVLT (7.0V) on this process is 93.

NOTE:
FIT = Failure in Time. 1 FIT = 1 failure per 1×10^9 device hours.

Failure Analysis

A. 1 oxide failure – 0.3 eV

5AC324

Device: 5AC324
Organization 24 Macrocells
Pinout: 40 Lead, 600 mil CERDIP and PDIP (44 Lead PLCC)
Die Size: 241 x 187 mils
Process: CHMOS IIIE, P429.3
Programming Voltage: 12.5V
Technology: CMOS

Table 1. Reliability Data

Package	Burn-In 48 Hours	125°C Dynamic Lifetest		
		168 Hours	500 Hours	1K Hours
CERDIP	3/2872	0/2869	0/298	0/298
TOTALS	3/2872 A	0/2869	0/298	0/298

Package	125°C High Voltage Dynamic Lifetest (7.0V)		
	48 Hours	168 Hours	500 Hours
CERDIP	0/113	0/113	0/113
TOTALS	0/113	0/113	0/113
Additional Readouts: CERDIP	0/113 @ 1K Hours High Voltage Lifetest		

Package	Data Retention Bake (CERDIP @ 250°C)		
	48 Hours	168 Hours	500 Hours
CERDIP	0/99	0/99	0/99
TOTALS	0/99	0/99	0/99
Additional Readouts: CERDIP	0/99 @ 1K Hours, 250°C Bake		

Table 2. Additional Qualification Stress Information

Package	85°C/85% Relative Humidity		
	168 Hours	500 Hours	1K Hours
PDIP	0/291	0/291	0/291
TOTALS	0/291	0/291	0/291

Package	Steam (121°C, 2 ATM)	
	96 Hours	168 Hours
PDIP	0/104	0/104
PLCC	0/481	0/481
TOTALS	0/585	0/584
Additional Readouts: PDIP	0/104 @ 500 Hours of Steam	
PLCC	0/333 @ 500 Hours of Steam	

Table 3. Failure Rate Predictions (5AC324-CERDIP-Fab 1)

125°C Actual Device Hours	Ea (eV)	Equivalent Hours 55°C	Equivalent Hours 70°C	# Fail	Fail Rate %/1K Hours (60% UCL) 55°C	Fail Rate %/1K Hours (60% UCL) 70°C
Gate Oxide						
5.92×10^5 1.13×10^5	0.3 B.I. 0.3 HVLT * VAF	2.73×10^6 4.85×10^7	1.84×10^6 3.26×10^7	0 0		
			Total 0.3 eV Failures =	0	0.0018	0.0027
5.92×10^5 9.94×10^4 1.13×10^5	0.5 B.I. 0.5 Bake 0.5 HVLT	7.57×10^6 7.23×10^7 1.44×10^6	3.90×10^6 3.34×10^7 7.44×10^5	0 0 0		
			Total 0.5 eV Failures =	0	0.0000	0.0000
5.92×10^5 9.94×10^4 1.13×10^5	0.6 B.I. 0.6 Bake 0.6 HVLT	1.26×10^7 2.70×10^8 2.40×10^6	5.69×10^6 1.07×10^8 1.09×10^6	0 0 0		
			Total 0.6 eV Failures =	0	0.0000	0.0000
5.92×10^5 1.13×10^5	1.0 B.I. 1.0 HVLT	9.68×10^7 1.85×10^7	2.57×10^7 4.90×10^6	0 0		
			Total 1.0 eV Failures =	0	0.0000	0.0000
			Combined Failure Rate: FITs:		0.0018 18	0.0027 27
48 Hour BI Infant Mortality = 3/2872 = 0.1045% = 1045 DPM						

Other Data (CERDIP)

		Temp with Theta Ja	
Theta Ja =	34°C/W	**Degree Kelvin**	
V_{CC} =	5.25V	T(55°C) =	355 K
I_{CC} @ 55°C =	150 mA	T(70°C) =	370 K
I_{CC} @ 70°C =	150 mA	T(125°C) =	420 K
I_{CC} @ 125°C =	125 mA	T(140°C) =	523 K

Boltzman's Constant = K = 8.62×10^{-5} eV/K

Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLT) Activation Energy	55°C	70°C
0.3	4.61	3.10
0.5	12.78	6.59
0.6	21.27	9.60
1.0	163.32	43.38

Bake (250°C) Activation Energy	55°C	70°C
0.3	n/a	n/a
0.5	727.61	335.91
0.6	2718.18	1075.15
1.0	n/a	n/a

Other Data (PDIP and PLCC)

		Temp with Theta Ja Degree Kelvin	
Theta Ja =	43°C/W		
V_{CC} =	5.25V	T(55°C) =	361.86 K
I_{CC} @ 55°C =	150 mA	T(70°C) =	376.86 K
I_{CC} @ 70°C =	150 mA	T(125°C) =	426.22 K
I_{CC} @ 125°C =	125 mA	T(140°C) =	413 K

Boltzman's Constant = K = 8.62×10^{-5} eV/K

Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLT)			Bake (140°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	4.27	2.91	0.3	n/a	n/a
0.5	11.23	5.94	0.5	38.00	17.54
0.6	18.22	8.48	0.6	78.65	31.11
1.0	126.20	35.25	1.0	n/a	n/a

Voltage Acceleration Factor (VAF) for HVLT (7.0V) on this process is 93.

NOTE:
FIT = Failure in Time. 1 FIT = 1 failure per 1×10^9 device hours.

Failure Analysis

A. 2 passivation crack – 0.5 eV
1 metal stringer – 0.5 eV

85C220

Device: 85C220
Organization 8 Macrocells
Pinout: 20 Lead, 300 mil CERDIP and PDIP; 20 Lead PLCC
Die Size: 89 x 90 mils
Process: CHMOS IIIE, P429.3
Programming Voltage: 12.5V
Technology: CMOS

Table 1. Reliability Data

Package	Burn-In 48 Hours	125°C Dynamic Lifetest		
		168 Hours	500 Hours	1K Hours
CERDIP	1/4383	1/4382	0/199	0/199
TOTALS	1/4383 A	1/4382 B	0/199	0/199
Additional Readouts: CERDIP 0/99 @ 2K Hours Lifetest				

Package	125°C High Voltage Dynamic Lifetest (7.0V)		
	48 Hours	168 Hours	500 Hours
CERDIP	0/103	0/103	0/103
TOTALS	0/103	0/103	0/103
Additional Readouts: CERDIP 0/103 @ 1K Hours High Voltage Lifetest			

Package	Data Retention Bake (CERDIP @ 250°C, PDIP @ 140°C)		
	48 Hours	168 Hours	500 Hours
CERDIP	0/105	0/105	0/105
TOTALS	0/105	0/105	0/105
Additional Readouts: CERDIP 0/105 @ 1K Hours, 250°C Bake			

Table 2. Additional Qualification Stress Information

Package	Temperature Cycling (Condition "C"—150°C to −65°C)		
	200 Cycles	500 Cycles	1K Cycles
CERDIP	0/49	0/49	0/49
PDIP	0/53	0/53	—
PLCC	0/56	0/56	0/56
TOTALS	0/158	0/158	0/105

Package	Thermal Shock (Condition "C"—150°C to −65°C)		
	50 Cycles	200 Cycles	500 Cycles
CERDIP	0/50	0/50	0/50
PDIP	0/52	0/52	0/52
PDIP	0/44	0/44	0/44
TOTALS	0/146	0/146	0/146

Package	85°C/85 % Relative Humidity		
	168 Hours	500 Hours	1K Hours
PDIP	0/207	0/207	0/207
TOTALS	0/207	0/207	0/207
Additional Readouts:			
PDIP	0/104 @ 2K Hours, 85°C, 85% Humidity		

Package	Steam (121°C, 2 ATM)	
	96 Hours	168 Hours
PDIP	0/106	0/106
PLCC	0/298	0/298
TOTALS	1/404 C	0/404
Additional Readouts:		
PDIP	0/106 @ 500 Hours of Steam	
PLCC	0/298 @ 500 Hours of Steam	

Table 3. Failure Rate Predictions (CERDIP 85C220)

125°C Actual Device Hours	Ea (eV)	Equivalent Hours		# Fall	Fall Rate %/1K Hours (60% UCL)	
		55°C	70°C		55°C	70°C
6.92×10^5 1.55×10^5	0.3 B.I. 0.3 HVLT * VAF	3.68×10^6 4.73×10^6	2.41×10^6 3.10×10^6	0 0		
Total 0.3 eV Failures =				0	0.0011	0.0017
6.92×10^5 1.05×10^5 1.55×10^5	0.5 B.I. 0.5 Bake 0.5 HVLT	1.12×10^7 7.67×10^7 2.50×10^6	5.53×10^6 3.54×10^7 1.24×10^6	1 0 0		
Total 0.5 eV Failures =				1	0.0022	0.0048
6.92×10^5 1.05×10^5 1.55×10^5	0.6 B.I. 0.6 Bake 0.6 HVLT	1.95×10^7 2.87×10^8 4.37×10^6	8.38×10^6 1.13×10^8 1.88×10^6	0 0 0		
Total 0.6 eV Failures =				0	0.0	0.0
6.92×10^5 1.55×10^5	1.0 B.I. 1.0 HVLT	1.81×10^8 4.04×10^7	4.42×10^7 9.89×10^6	0 0		
Total 1.0 eV Failures =				0	0.0	0.0
Combined Failure Rate: FITs:					0.0033 33	0.0065 65
48 Hour BI Infant Mortality = 2/4404 = 0.0454 = 454 DPM						

3

Other Data (CERDIP)

		Temp with Theta Ja Degree Kelvin	
Theta Ja =	76°C/W		
V_{CC} =	5.25V	T(55°C) =	344 K
I_{CC} @ 55°C =	40 mA	T(70°C) =	359 K
I_{CC} @ 70°C =	40 mA	T(125°C) =	412 K
I_{CC} @ 125°C =	35 mA	T(140°C) =	523 K

Boltzman's Constant = K = 8.62×10^{-5} eV/K

Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLT)			Bake (250°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	5.31	3.48	0.3	n/a	n/a
0.5	16.15	7.99	0.5	727.61	335.91
0.6	28.18	12.10	0.6	2718.18	1075.15
1.0	260.95	63.80	1.0	n/a	n/a

Other Data (PDIP and PLCC)

		Temp with Theta Ja Degree Kelvin	
Theta Ja =	90°C/W	T(55°C) =	346.90 K
V_{CC} =	5.25V	T(70°C) =	361.90 K
I_{CC} @ 55°C =	40 mA	T(125°C) =	414.54 K
I_{CC} @ 70°C =	40 mA	T(140°C) =	413 K
I_{CC} @ 125°C =	35 mA		

Boltzman's Constant = K = 8.62×10^{-5} eV/K

Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLT)			Bake (140°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	5.13	3.39	0.3	n/a	n/a
0.5	15.28	7.64	0.5	38.00	17.54
0.6	26.36	11.48	0.6	78.65	31.11
1.0	233.53	58.44	1.0	n/a	n/a

Voltage Acceleration Factor (VAF) for HVLT (7.0V) on this process is 93.

NOTE:
FIT = Failure in Time. 1 FIT = 1 failure per 1×10^9 device hours.

Failure Analysis

A. 1 ISB - metal stringer – 0.5 eV
   1 verify - in analysis – TBD
B. 1 leakage - assembly defect – 0.5 eV

85C508

Device: 85C508
Organization Decoder/Latch PLD
Pinout: 28 Lead, 300 mil CERDIP and PDIP; 28 Lead PLCC
Die Size: 65 x 98 mils
Process: CHMOS IIIE, P429.5
Programming Voltage: 12.5V
Technology: CMOS

Table 1. Reliability Data

Package	Burn-In 48 Hours	125°C Dynamic Lifetest		
		168 Hours	500 Hours	1K Hours
CERDIP	1/3387	0/3358	0/201	0/198
TOTALS	1/3387 A	0/3358	0/201	0/198
Addtional Readouts: CERDIP	0/102 @ 2K Hours Lifetest			

Package	125°C High Voltage Dynamic Lifetest (7.0V)		
	48 Hours	168 Hours	500 Hours
CERDIP	0/205	0/205	0/205
TOTALS	0/205	0/205	0/205
Additional Readouts: CERDIP	0/102 @ 1.5K Hours High Voltage Lifetest		

Package	Data Retention Bake (CERDIP @ 250°C, PDIP @ 140°C)		
	48 Hours	168 Hours	500 Hours
CERDIP	0/99	0/99	0/99
TOTALS	0/99	0/99	0/99

Table 2. Additional Qualification Stress Information

Package	Temperature Cycling (Condition "C"—150°C to −65°C)		
	200 Cycles	500 Cycles	1K Cycles
CERDIP	0/364	0/364	0/358
TOTALS	0/364	0/364	0/358

Package	Thermal Shock (Condition "C"—150°C to −65°C)		
	50 Cycles	200 Cycles	500 Cycles
CERDIP	0/353	0/353	0/353
TOTALS	0/353	0/353	0/353

Package	Steam (121°C, 2 ATM)	
	96 Hours	168 Hours
PDIP	0/50	0/50
PLCC	0/50	0/50
TOTALS	0/100	0/100

Table 3. Failure Rate Predictions (CERDIP 85C508)

125°C Actual Device Hours	Ea (eV)	Equivalent Hours 55°C	Equivalent Hours 70°C	# Fail	Fail Rate %/1K Hours (60% UCL) 55°C	Fail Rate %/1K Hours (60% UCL) 70°C
6.71×10^5 2.06×10^5	0.3 B.I. 0.3 HVLT * VAF	3.94×10^6 1.13×10^8	2.52×10^6 7.21×10^7	0 0		
		Total 0.3 eV Failures =		0	0.0008	0.0012
6.71×10^5 5.04×10^4 2.06×10^5	0.5 B.I. 0.5 Bake 0.5 HVLT	1.28×10^7 3.67×10^7 3.94×10^6	6.10×10^6 1.69×10^7 1.87×10^6	0 — 0		
		Total 0.5 eV Failures =		0	0.0	0.0
6.71×10^5 5.04×10^4 2.06×10^5	0.6 B.I. 0.6 Bake 0.6 HVLT	2.32×10^7 1.37×10^8 7.12×10^6	9.49×10^6 5.42×10^7 2.92×10^6	0 — 0		
		Total 0.6 eV Failures =		0	0.0	0.0
6.71×10^5 2.06×10^5	1.0 B.I. 1.0 HVLT	2.46×10^8 7.55×10^7	5.56×10^7 1.71×10^7	0 0		
		Total 1.0 eV Failures =		0	0.0	0.0
			Combined Failure Rate: FITs:		0.0008 8	0.0012 12
48 Hour BI Infant Mortality = 1/3387 = 0.0295 = 295 DPM						

Other Data (CERDIP)

		Temp with Theta Ja Degree Kelvin	
Theta Ja =	83°C/W	T(55°C) =	335 K
V_{CC} =	5.25V	T(70°C) =	350 K
I_{CC} @ 55°C =	15 mA	T(125°C) =	403 K
I_{CC} @ 70°C =	15 mA	T(140°C) =	523 K
I_{CC} @ 125°C =	12 mA		

Boltzman's Constant = K = 8.62×10^{-5} eV/K

Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLT) Activation Energy	55°C	70°C	Bake (250°C) Activation Energy	55°C	70°C
0.3	5.88	3.76	0.3	n/a	n/a
0.5	19.15	9.10	0.5	727.61	335.91
0.6	34.55	14.16	0.6	2718.18	1075.15
1.0	366.55	82.83	1.0	n/a	n/a

Other Data (PDIP and PLCC)

		Temp with Theta Ja Degree Kelvin	
Theta Ja =	100°C/W		
V_{CC} =	5.25V	T(55°C) =	335.53 K
I_{CC} @ 55°C =	15 mA	T(70°C) =	349.53 K
I_{CC} @ 70°C =	15 mA	T(125°C) =	403.23 K
I_{CC} @ 125°C =	12 mA	T(140°C) =	413 K

Boltzman's Constant = K = 8.62×10^{-5} eV/K

Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLT)			Bake (140°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	5.77	3.71	0.3	n/a	n/a
0.5	18.56	8.87	0.5	38.00	17.54
0.6	33.29	13.73	0.6	78.65	31.11
1.0	344.53	78.76	1.0	n/a	n/a

Voltage Acceleration Factor (VAF) for HVLT (7.0V) on this process is 93.

NOTE:
FIT = Failure in Time. 1 FIT = 1 failure per 1×10^9 device hours.

Failure Analysis

A. Oxide Defect – 0.3 eV

3

85C960

Device: 85C960
Organization 80950 K-Series PLD
Pinout: 28 Lead, 300 mil CERDIP and PDIP; 28 Lead PLCC
Die Size: 77 x 122 mils
Process: CHMOS IIIE, P429.5
Programming Voltage: 12.5V
Technology: CMOS

Table 1. Reliability Data

Package	Burn-In 48 Hours	125°C Dynamic Lifetest		
		168 Hours	500 Hours	1K Hours
CERDIP	1/2176	0/2173	0/455	0/400
TOTALS	1/2176 A	0/2173	0/455	0/400
Addtional Readouts: CERDIP	0/200 @ 2K Hours Lifetest			

Package	125°C High Voltage Dynamic Lifetest (7.0V)		
	48 Hours	168 Hours	500 Hours
CERDIP	1/400	0/398	0/398
TOTALS	1/400 B	0/398	0/398
Additional Readouts: CERDIP	0/398 @ 1K Hours High Voltage Lifetest		

Table 2. Additional Qualification Stress Information

Package	Temperature Cycling (Condition "C"—150°C to −65°C)		
	200 Cycles	500 Cycles	1K Cycles
CERDIP	0/25	0/25	0/25
PDIP	0/25	0/25	0/25
PLCC	0/25	0/25	0/25
TOTALS	0/75	0/75	0/50

Package	Thermal Shock (Condition "C"—150°C to −65°C)		
	50 Cycles	200 Cycles	500 Cycles
CERDIP	0/25	0/25	0/25
PDIP	0/25	0/25	0/24
PLCC	0/25	0/25	1/25
TOTALS	0/75	0/75	1/49 C
CERDIP	0/353	0/353	0/353

Package	Steam (121°C, 2 ATM)	
	96 Hours	168 Hours
PDIP	0/100	0/100
PLCC	1/100	0/100

Additional Readouts:

PLCC	0/100 @ 500 Hours of Steam
PDIP	0/50 @ 500 Hours of Steam

3

Table 3. Failure Rate Predictions (CERDIP 85C960)

125°C Actual Device Hours	Ea (eV)	Equivalent Hours		# Fail	Fail Rate %/1K Hours (60% UCL)	
		55°C	70°C		55°C	70°C
8.12×10^5 3.98×10^5	0.3 B.I. 0.3 HVLT * VAF	5.06×10^6 1.43×10^6	3.21×10^6 9.05×10^6	0 0		
		Total 0.3 eV Failures =		0	0.0047	0.0075
8.12×10^5 — 3.98×10^5	0.5 B.I. 0.5 Bake 0.5 HVLT	1.71×10^7 — 8.40×10^6	8.04×10^6 — 3.94×10^6	0 — 1		
		Total 0.5 eV Failures =		0	0.0079	0.0169
8.12×10^5 — 3.98×10^5	0.6 B.I. 0.6 Bake 0.6 HVLT	3.15×10^7 — 1.54×10^7	1.27×10^7 — 6.23×10^6	0 — 0		
		Total 0.6 eV Failures =		0	0.0	0.0
8.12×10^5 3.98×10^5	1.0 B.I. 1.0 HVLT	3.61×10^8 1.77×10^8	7.95×10^7 3.90×10^7	0 0		
		Total 1.0 eV Failures =		0	0.0	0.0
			Combined Failure Rate: FITs:		0.126 126	0.0244 244
48 Hour BI Infant Mortality = 1/2176 = 0.0460 = 460 DPM						

Other Data (CERDIP)

		Temp with Theta Ja Degree Kelvin	
Theta Ja =	66.5°C/W		
V_{CC} =	5.25V	T(55°C) =	332 K
I_{CC} @ 55°C =	10 mA	T(70°C) =	347 K
I_{CC} @ 70°C =	10 mA	T(125°C) =	402 K
I_{CC} @ 125°C =	10 mA	T(140°C) =	523 K

Boltzman's Constant = K = 8.62×10^{-5} eV/K

Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLT)			Bake (250°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	6.23	3.96	0.3	n/a	n/a
0.5	21.09	9.89	0.5	727.61	335.91
0.6	38.82	15.65	0.6	2718.18	1075.15
1.0	444.99	97.90	1.0	n/a	n/a

Other Data (PDIP and PLCC)

		Temp with Theta Ja Degree Kelvin	
Theta Ja =	81°C/W		
V_{CC} =	5.25V	T(55°C) =	332.25 K
I_{CC} @ 55°C =	10 mA	T(70°C) =	347.25 K
I_{CC} @ 70°C =	10 mA	T(125°C) =	402.25 K
I_{CC} @ 125°C =	10 mA	T(140°C) =	413 K

Boltzman's Constant = K = 8.62×10^{-5} eV/K

Thermal Acceleration Factors—

Lifetest (LT) and High Voltage Lifetest (HVLT)			Bake (140°C)		
Activation Energy	55°C	70°C	Activation Energy	55°C	70°C
0.3	6.18	3.93	0.3	n/a	n/a
0.5	20.83	9.80	0.5	38.00	17.54
0.6	38.23	15.47	0.6	78.65	31.11
1.0	433.79	96.08	1.0	n/a	n/a

Voltage Acceleration Factor (VAF) for HVLT (7.0V) on this process is 93.

NOTE:
FIT = Failure in Time. 1 FIT = 1 failure per 1×10^9 device hours.

Failure Analysis

A. Fab Defect – 0.5 eV
B. Metal Stringer – 0.5 eV
C. Polysilicon Defect – 0.5 eV

3

APPENDIX A
FAILURE RATE CALCULATIONS FOR 60% UPPER CONFIDENCE LEVEL

Step 1. Collect burn-in and lifetest data for each lot after 48 hours of burn-in through lifetest for each lot.

Step 2. Determine the failure mechanism and assign an activation energy (E_A) for each failure, except those occurring during the first 48 hrs. (See Table 1 below.)

Table 1. Failure Mechanism Activation Energies Relevant to EPROMs

Failure Mode	Activation Energy
Defective bit charge gain/loss	0.6 eV
Oxide breakdown	0.3 eV
Silicon defects	0.3 eV
Fab/Assembly Defects	0.5 eV
Contamination	1.0–1.2 eV
Intrinsic charge loss	1.4 eV

Step 3. Calculate the total number of device hours accumulated beyond 48 hours of burn-in. (**Note:** 48 hour burn-in results measure infant mortality and are not included in the failure rate calculation.)

Example: 125°C Burn-In/Lifetest for a 2 lot sample

$$\frac{\text{\# failures}}{\text{total \# devices}}$$

	48 Hours	168 Hours	500 Hours	1K Hours	2K Hours
Lot #1	0/1000	1/1000	0/999	0/998	0/994
Lot #2	0/221	0/201	1/201	1/100	0/99
Totals	0/1221	1/1201	1/1200	1/1098	0/1093

Device Hours = (Number of Devices) (Number of Hours)

Total Device Hours = 1201 (168 hrs − 48 hrs) + 1200 (500 hrs − 168 hrs) + 1098 (1000 hrs − 500 hrs) + 1093 (2000 hrs − 1000 hrs)

= 1201 (120 hrs) + 1200 (332 hrs) + 1098 (500 hrs) + 1093 (1000 hrs)

= 2.185×10^6 Device Hours

Step 4. Use E_A tables to find the equivalent device hours at a desired temperature for each activation energy (failure mechanism), or use the Arrhenius relation.

$$R = A \exp\left[\frac{-E_A}{KT}\right]$$

K = 8.617×10^{-5} eV/°K (Boltzmann's constant)
E_A = activation energy
A = proportionality constant
T = temperature in Kelvin
R = mean rate to failure

$$\frac{R_1}{R_2} = \frac{A_1 \exp\left[\frac{-E_A}{KT_1}\right]}{A_2 \exp\left[\frac{-E_A}{KT_2}\right]} = \exp\left[\left(\frac{E_A}{K}\right)\left(\frac{1}{T_2} - \frac{1}{T_1}\right)\right]$$

Where $A_1 = A_2 = A$ for the same failure mechanism (i.e., same E_A)

Where R_1 and R_2 are rates for a normal operating temperature and an elevated temperature respectively.

$$R_1 = R_2 \times \exp\left[\left(\frac{E_A}{K}\right)\left(\frac{1}{T_2} - \frac{1}{T_1}\right)\right]$$

However, since rate (R) has the units 1/time, we can think in terms of time to one failure or MTBF.

Thus:

$$R_1 = \frac{1}{t_1} \text{ where } t_1 = \text{MTBF at some temperature } T_1$$

and:

$$R_2 = \frac{1}{t_2} \text{ where } t_2 = \text{MTBF at some temperature } T_2$$

Thus the Arrhenius Relation becomes:

$$\frac{1}{t_1} = \frac{1}{t_2} \times \exp\left[\frac{E_A}{K}\left(\frac{1}{T_2} - \frac{1}{T_1}\right)\right]$$

or:

$$t_1 = \exp\left[\frac{E_A}{K}\left(\frac{1}{T_1} - \frac{1}{T_2}\right)\right] \times t_2$$

We then define the Acceleration Factor as:

$$\text{A.F.} = \frac{t_1}{t_2} = \exp\left[\frac{E_A}{K}\left(\frac{1}{T_1} - \frac{1}{T_2}\right)\right]$$

For example: For $E_A = 0.6$ eV, $T_2 = 398°K$, $T_1 = 328°K$

$$t_1 = 41.7\, t_2$$

Therefore, one hour at 125° is equivalent to 41.7 hours at 55°C for a failure mechanism of activation energy E_A = 0.6 eV. Then 41.7 is the thermal acceleration factor for time.

NOTE:
The Arrhenius Plot (Figure 2) is simply ln (Acceleration Factor) vs. 1/Temperature normalized for an MTBF (t_2) of one hour at 250°C (T_2). This plot can also be used to determine the acceleration factor between two temperatures (other than 250°C).

3

For example: For a 0.3 eV failure at 125°C, the acceleration factor is 8.1 relative to a 0.3 eV failure at 250°C. For a 0.3 eV failure at 25°C, the acceleration factor is 152 relative to 250°C. Therefore, the acceleration factor between 125°C and 25°C is:

$$\text{A.F.} = \frac{t_1}{t_2} = \frac{152}{8.1} = 18.7$$

Step 5. Organize the burn-in/lifetest data by E_A, Total Device Hours at the burn-in/lifetest temperature T_2, Thermal Acceleration Factors for each failure mechanism (E_A), Number of Failures for each failure mechanism, and the calculated equivalent device hours at the desired operating temperature T_1.

NOTE:
The rise in junction temperature due to the thermal resistivity of the package (θ_{JA}) must be added to the desired and actual burn-in/lifetest temperatures.

$$T_{test} = T_J + T_{Ambient} = \theta_{JA}\,(IV \text{ @ } T_{Ambient}) + T_{Ambient}$$

E_A (eV)	Total Device Hours @ T_2	Acceleration Factors	# Fail	Equivalent Hours @ T_1
0.3	T.D.H.	X	N_1	X(T.D.H.)
0.6	T.D.H.	Y	N_2	Y(T.D.H.)
1.0	T.D.H.	Z	N_3	Z(T.D.H.)

The failure rates for individual failure mechanisms and the total combined failure rate can be predicted using the data table and the following formula:

$$\%\ \text{fail/1K hours} = \frac{\chi^2\,(n, \alpha)}{2T}\left(10^5\right)$$

Where χ^2 (n, α) is the value of the chi-squared distribution for n degrees of freedom and confidence level of α. T is the total equivalent device hours at T_1. The total combined failure rate is just the sum of the individual failure rates for each failure mechanism.

For a 60% VCL, the above formula converts to the following:

# Failures	% Fail/1K Hours (60% UCL)
0	$0.915 \times 10^5/T$
1	$2.02 \times 10^5/T$
2	$3.105 \times 10^5/T$
3	$4.17 \times 10^5/T$

$$3 < \# < 15\left[\frac{1.049\,(\#\text{ failures for a particular } E_A) + 1.0305}{T}\right]\left[10^5\right]$$

$$> 15\,\frac{[0.2533 + \sqrt{(4 \times \#\text{ Failed}) + 3}]^2}{4T}\left[10^5\right]$$

Example 1:

Assume for this example, that I_{CC} active is 57 mA at $T_{Ambient}$ = 125°C and I_{CC} active is 60 mA at $T_{Ambient}$ = 55°C.

Also assume that θ_{JA} = 35°C/W.

Then,

T_2 = (35°C/W) (57 mA) (5V) + 125°C
≈ 135°C = 408°K

T_1 = (35°C/W) (60 mA) (5V) + 55°C
≈ 65°C = 338°K

E_A (eV)	Actual Device Hours @ 125°C	Acceleration Factors for 135°C to 65°C	Equivalent Hours at 55°C	# Fail	55°C %Fail/ 1K Hrs
0.3	2.185×10^6	5.85	1.278×10^7	0	0.0081
0.6	2.185×10^6	34.18	7.468×10^7	2	0.0042
1.0	2.185×10^6	359.93	7.864×10^8	1	0.0003
				Total Combined Failure Rate =	0.0126 = 126 FITs

Example 2:

Assume that an additional lot of 800 HMOS*IIE devices is burned in using a 6.5V lifetest. Using Table 2 below, a voltage acceleration factor of 55 results from a 20% overstress (5.5V to 6.5V).

	48 Hours	168 Hours	500 Hours
Lot #3	0/800	1/800	0/799

Device Hours = 800 (48 hrs − 0 hrs) + 800 (168 hrs − 48 hrs) + 799 (500 hrs − 168 hrs)
= 3.997×10^5

Table 2. Time-Dependent Oxide Failure Accelerations

Type	Supply Voltage (Volts)	Oxide Thickness (Å)	Operating Stress (MV/cm)	Lifetest Stress Voltage Relative to 5.25V			
				5.5	6.0	6.5	7.0
CHMOS II E	5	400	1.25	1.6	7.5	55	422
CHMOS III E	5	235	2.15	1.9	7.0	26	93

ASSUMES:
1. Failure rate calculations use the appropriate acceleration factor for stress voltage and maximum operating voltage (conservative).
2. Reference [10] E. Nelson Anolick.

Since this voltage accelerated stress is used to predict an oxide breakdown failure rate, the 5.5V burn-in/lifetest 55°C equivalent hours for E_A = 0.3 eV are added to the 6.5V burn-in/lifetest 55°C equivalent hours as follows:

125°C Burn-In/Lifetest	E_A (eV)	Actual Device Hours @ 125°C	Acceleration Factors for 135°C to 65°C	Equivalent Hours 55°C
5.5V	0.3	2.185×10^6	5.85	1.278×10^7
6.5V	0.3	3.997×10^5	(5.85 x 55)	1.286×10^8
			Total Equivalent E_A = 0.3 eV Device Hours =	1.414×10^8

The following failure rate predictions include the total equivalent 55°C, E_A = 0.3 eV device hours found above:

E_A (eV)	Actual Device Hours @ 125°C	Acceleration Factors for 135°C at 65°C	Equivalent Hours @55°C	# Fail	55°C % Fail/ 1K Hours
0.3	2.185×10^6	5.85	—	—	—
0.3 + 55(1)	3.997×10^5	(5.85 x 55)	1.414×10^8	1	0.0015
0.6	2.185×10^6	34.18	7.468×10^7	2	0.0042
1.0	2.185×10^6	359.93	7.864×10^8	1	0.0003
				Total Combined Failure Rate =	0.0060
				=	60 FITs

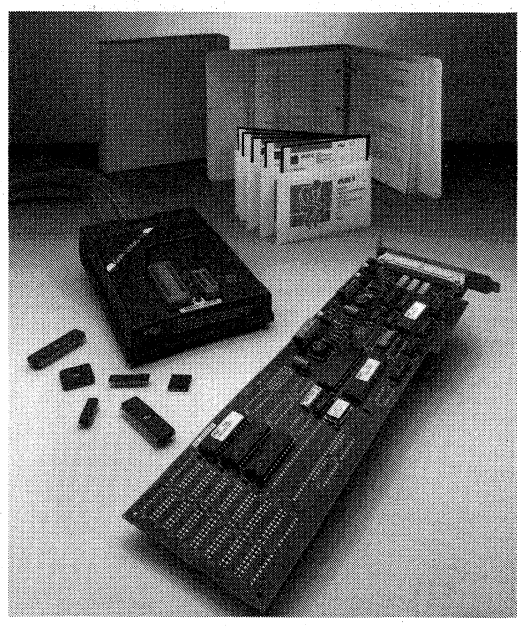
NOTES:
1. The notation 0.3 + 55 is used to show that 6.5V and 5.5V burn-in/lifetest equivalent hours have been combined.
2. Additional information on calculating failure rates is contained in the April 2, 1984 International Reliability Physics Symposium editorial entitled "Calculating Failure Rates from Stress Data" by Robert M. Alexander.
3. 1 FIT = 1 Failure In Time = 0.0001%/1K Hours.

Development Support Tools 4

iPLDS II
INTEL PROGRAMMABLE LOGIC DEVELOPMENT SYSTEM VERSION II

- **Hardware and Software Necessary to Turn Design Concepts into Functional Erasable Programmable Logic Devices (EPLDs)**
- **Menu-driven Software with On-line Help Messages for All Stages of the Design Process**
- **iUP-PC Hardware Programs Intel EPLD's, EPROM's, Peripherals, and Microcontrollers with one PC-based System**
- **All Equipment Interfaces with the IBM PC/XT*, PC/AT*, and True Compatibles**
- **iPLS II (Software Only) Available on DOS and Workstation Platforms.**
- **JEDEC Standard Design File, Part Utilization Report, Minimized Equation File, and Compiler Error File All Available as Outputs**
- **Supports a Variety of Input Methods:**
  - **Schematic Entry**
    - **— TTL Library**
    - **— EPLD Primitives Library**
  - **Text Editor Entry**
    - **— State Machine**
    - **— Boolean Equations**
- **Macro Expander Accepts TTL, and User-Defined Macros and Expands Them into Equivalent EPLD Primitives**
- **Espresso** Minimizer Reduces Logic Equations to Least Number of Product Terms**
- **Supports All Intel EPLD's Including the 85C220, 85C224, 85C060, and 85C090**

4



290134-1

iPLDS II Components Picture

*IBM PC/XT, PC/AT are registered trademarks of International Business Machines Corporation.

**ESPRESSO is copyrighted by the University of California at Berkeley and is used with permission.

This product is manufactured by Intel Puerto Rico, Inc.

October 1990
Order Number: 290134-007

INTRODUCTION

Intel's Programmable Logic Development System II (iPLDS II) is a powerful set of tools for transforming a logic design into customized silicon. The system provides design entry, logic compilation, and device programming capability on a desktop using an IBM PC/XT, PC/AT, or compatible.

SUMMARY OF PROGRAMMABLE LOGIC DESIGN

When performing a programmable logic design on a CAD system, the design must first be entered using one of a variety of entry methods. These methods typically include schematic capture or Boolean equation entry using a standard text editor. Other entry methods include netlist entry, whereby a hand drawn schematic can be entered in a node-by-node fashion, or state machine entry in a text or graphical mode.

Once the design has been entered into the CAD package, several processing steps may occur. The design is usually translated into a format usable by the software, logic reduction may be performed, and, finally, some form of programming file can be produced. Most CAD packages also produce documentation of the minimization and device fitting results, including the final pin assignments.

Once the programming file has been generated, the design can be transferred into silicon in a programming manner similar to that used for EPROMs.

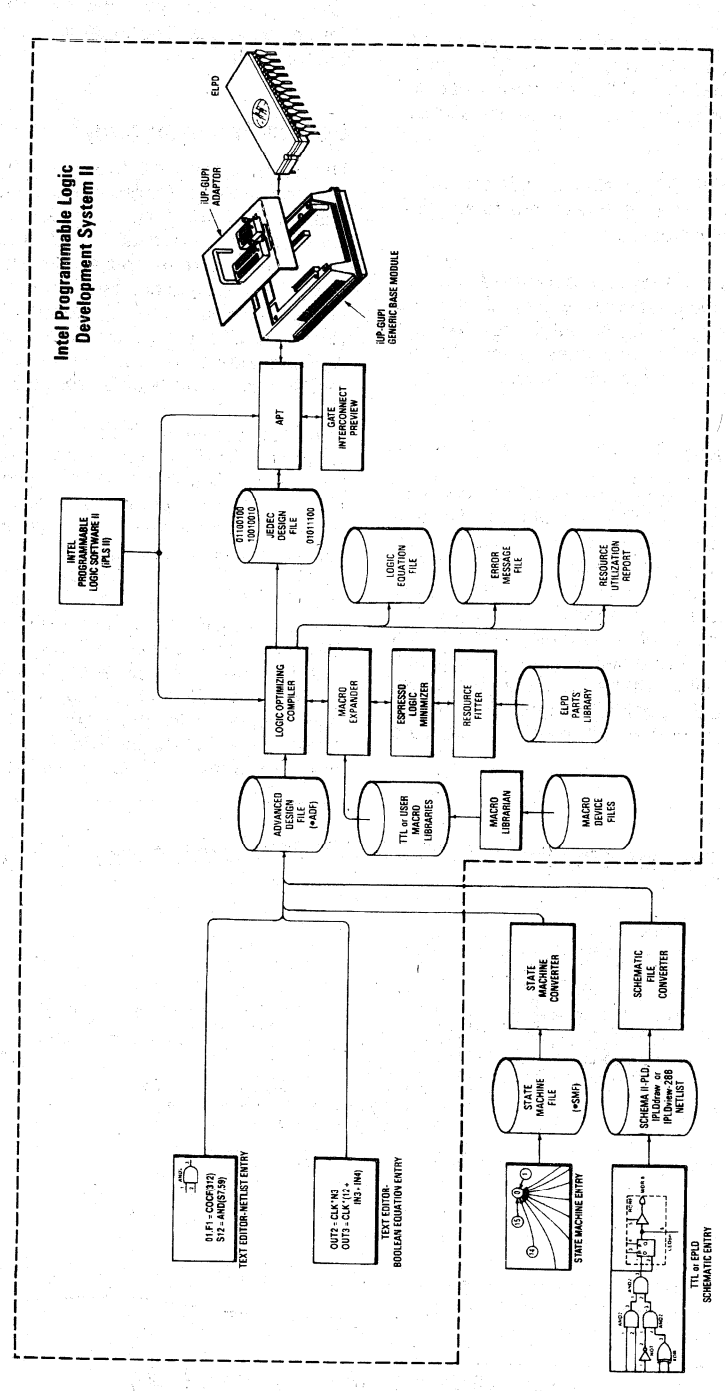
FUNCTIONAL DESCRIPTION OF iPLDS II

All of the design entry methods with the exception of graphic state machine entry are supported by the iPLDS II software. iPLDS II supports netlist and Boolean equation entry using any standard text editor. State machine software and schematic capture libraries are also available from Intel as optional entry methods. Depending on the entry format used, the design may require translation into Advanced Design File (ADF) format. Once the design is in ADF form, the Logic Optimizing Compiler expands any macros, minimizes all equations, and fits the design into a device-specific JEDEC Design File. The JEDEC Design File is programmed into the EPLD by APT programming software using the iUP-PC hardware. Thus, the circuit design is transformed into an operating EPLD on one workstation.

Intel Programmable Logic Software II (iPLS II) is composed of four functional modules: design entry, netlist conversion, file compilation and device programming. This is the same software as that included in the iPLDS II Development System.

Design Entry

Design entry is typically accomplished by creating an ADF using an ASCII text editor, or by using a schematic capture package.


Intel Programmable Logic Development System II
290134-2
ELPD
iUP-GUPI ADAPTOR
iUP-GUPI GENERIC BASE MODULE
INTEL PROGRAMMABLE LOGIC SOFTWARE II (iPLS II)
APT
GATE INTERCONNECT PREVIEW
JEDEC DESIGN FILE
LOGIC EQUATION FILE
ERROR MESSAGE FILE
RESOURCE UTILIZATION REPORT
LOGIC OPTIMIZING COMPILER
MACRO EXPANDER
ESPRESSO LOGIC MINIMIZER
RESOURCE FITTER
ELPD PARTS LIBRARY
ADVANCED DESIGN FILE (*.ADF)
TTL or USER MACRO LIBRARIES
MACRO LIBRARIAN
MACRO DEVICE FILES
STATE MACHINE CONVERTER
SCHEMATIC FILE CONVERTER
STATE MACHINE FILE (*.SMF)
SCHEMA II-PLD IPLDraw or IPLDview 288 NETLIST
TEXT EDITOR-NETLIST ENTRY
TEXT EDITOR-BOOLEAN EQUATION ENTRY
STATE MACHINE ENTRY
TTL or EPLD SCHEMATIC ENTRY

Netlist Conversion

If schematic capture of state machine entry is used, the design must be converted into an ADF format. The optional SCHEMA III-PLD schematic capture package is a low-cost way to enter schematic designs.

IPLDview-286 and IPLDdraw also provide schematic entry for EPLD designs. Both have paths to A.C. timing simulation of fitted designs.

State machine entry may be performed via the iSTATE software and a standard text editor. iSTATE is an option for DOS platforms but is included with iPLS II for workstation packages. Refer to the iSTATE data sheet for more information.

File Compilation

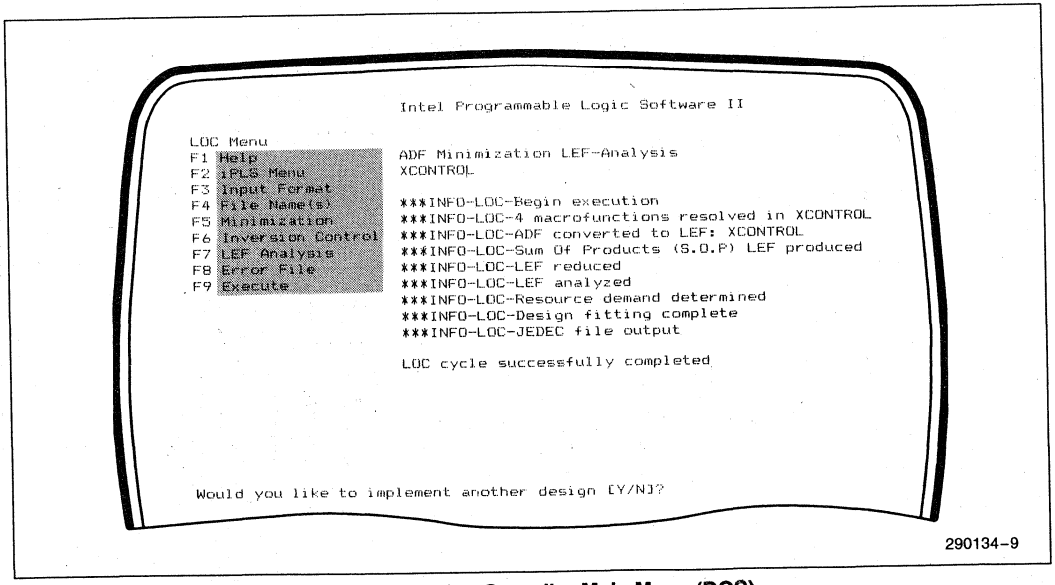
File compilation is performed by the LOC (Logic Optimizing Compiler). The LOC accepts an ADF and converts it into an industry standard JEDEC file which is used to program the device. As a part of this process, the LOC expands TTL macros into equivalent EPLD logic, minimizes the logic equations using the Espresso algorithm, and maps the network and logic equations into a cell map for the selected device. The final output of the LOC is a JEDEC Design File. The JEDEC Design File describes the design for the designated EPLD in JEDEC standard format.

For designs using the 5AC312 or 5AC324, iPLS II utilizes proprietary algorithms to efficiently use the device resources. The Fitter in iPLS II optimizes fitting for all devices.

Device Programming

The programming hardware is controlled by APT (Advanced Programming Tool) or LPS (Logic Programmer Software). These products take the JEDEC file produced by the LOC and program it into the device. They can also read a programmed device or verify that a device has been programmed correctly. LPS provides a graphical interface for standard devices and is available with the DOS product only. APT provides a command line interface for all Intel EPLDs and supports all current devices.

The Intel Universal Programmer for the Personal Computer (iUP-PC) is a versatile programming solution in a PC-based system. Installed in an IBM PC/XT, PC/AT or compatible host, the iUP-PC emulates the performance of the standalone Intel iUP-200A Universal Programmers. As such, it supports the iUP Generic Universal Programmer Interface (iUP-GUPI). With the appropriate socket adapters for the iUP-GUPI, the iUP-PC supports all Intel EPLDs. Future EPLDs will be supported by new GUPI adapters or adapter upgrades. Many other Intel devices—EPROMs, and microcontrollers—are also supported by the GUPI. The iUP-PC is controlled by APT/LPS or iPPS (Intel PROM Programmer Software). iPLDS II includes the iUP-PC,



Logic Optimizing Compiler Main Menu (DOS)

which contains the iPPS, PCPP programming card, interconnect cable, and the GUPI base. GUPI adapters are available separately.

iPLS II SOFTWARE

The Intel Programmable Logic Software II (iPLS II) has many options and enhancements for implementing a logic design. iPLS II accommodates a wide variety of design input methods. Schematics, state machines or Boolean equations may all be used provided the proper formats and convertors are implemented as needed. No matter what method is chosen, the Logic Optimizing Compiler will minimize and fit the design during compilation. iPLS II also contains APT (and LPS for DOS) programming software.

I. Design Input

The entire spectrum of design input methods is available to the logic designer in iPLS II. Everything from TTL schematics to Boolean equations are accepted and processed by the LOC.

A. SCHEMATIC ENTRY—MACRO SYMBOLS

Users can design using schematic capture software and macro symbols. Two types of symbols are available: TTL macro symbols, and EPLD custom macro symbols.

With TTL macro symbols designers can use familiar 74-series block symbols to create the schematic for their design. The TTL symbols are eventually netlisted and expanded into ADF Network and Equation entries for processing by the LOC. Only TTL symbols included in the symbol library for the respective schematic capture package are available for use. TTL symbols not supported can be created by the user and matched with user-created macro definitions. The iPLS II Macro Librarian can be used to create new macro definition libraries.

EPLD custom macro symbols are common groups of Intel design primitives preconnected in groups of 2, 4, 6, or 8 elements. EPLD custom macros speed design using the Intel design primitive symbols. These symbols are netlisted and expanded into individual ADF primitive entries for processing by the LOC.

B. SCHEMATIC ENTRY—INTEL SYMBOL LIBRARY

Lower-level design primitives are also available for schematic capture software. Design primitives give the designer greater control over the individual aspects of a design than TTL or EPLD custom macro symbols. Intel design primitive symbols are netlisted into ADF entries on a one-for-one basis for processing by the LOC.

C. TEXT EDITOR ENTRY

Designers who are familiar with the logic primitives and the Advanced Design File format can directly enter ADFs with a standard text editor. The bulk of the design entry can be accomplished using Boolean Equations obtained from a Karnaugh map or truth table. Hence, the need for conversion to gates is eliminated. This method of entry is useful for sub-circuits that will be incorporated into larger designs.

D. STATE MACHINE ENTRY

In the past, state diagrams or flowcharts (ASM charts) were merely abstractions used to obtain the logic equations necessary to implement TTL designs. With the advent of the iPLS II state machine convertor (iSTATE), this is no longer the case. Using an IF THEN / ELSE format, the designer may enter the state machine description without having to extract the logic and convert the equations into TTL components. The state machine to Boolean logic conversion is handled by the state machine convertor, provided the input file adheres to the specified State Machine File (SMF) format. iSTATE is an option for DOS packages but is included with iPLS II for workstation packages. Refer to the iSTATE Data Sheet for more information.

Summary of Optional Entry Methods:

SCHEMA III-PLD Schematic Capture

1. TTL Macro Library
2. EPLD Custom Macro Library
3. EPLD Primitive Symbols

IPLDview-286, IPLDdraw Schematic Capture

1. TTL Macro Library
2. EPLD Primitive Symbols
3. A.C. Timing Simulation (IPLDview-286)

State Machines

1. State Machine File (SMF) format used
2. Optional state machine convertor used in LOC (Convertor contained in iSTATE)

II. Logic File Compilation

Before programming the part, the designer must compile the input design file into a JEDEC standard file. This function is performed by the Logic Optimizing Compiler.

LOGIC OPTIMIZING COMPILER (LOC)

Once the input file is in Advanced Design File (ADF) format, the LOC will compile it into a device-specific JEDEC Design File. The first phase of this compilation is performed by the MACRO EXPANDER. The Macro Expander expands Intel or TTL macros into equivalent EPLD equations. The second phase is performed by the ESPRESSO MINIMIZER. The minimizer reduces all the logic equations to their simplest form using the ESPRESSO II-MV algorithm. The final phase of compilation is performed by the FITTER. The Fitter creates a cell map of the minimized equations according to the resources available within the specified device.

MACRO RESOLVER

The input design file is initially passed through the MACRO RESOLVER. The Macro Resolver searches the file for any non-EPLD network elements. If found, the Resolver then searches the User Libraries and TTL Library for the unidentified element. Once the element is located, the design file element is replaced by the equivalent EPLD primitive implementation found in the library. Having the Resolver search the User Libraries allows the user to create his own macros. User macro files are created with a standard ASCII text editor and are stored in libraries by the iPLS II Macro Librarian.

ESPRESSO MINIMIZER

The minimization in the LOC is performed by the ESPRESSO II-MV MINIMIZER. Developed by the University of California at Berkeley, the ESPRESSO II-MV algorithm is regarded by many as being the best minimization method available. ESPRESSO II-MV uses DeMorgan's and other logic theorems to reduce the equations to the least number of product terms possible. Since product terms are the key variable in the EPLD architecture, the ESPRESSO II-MV Minimizer provides the simplest equations possible. As a result, the success rate for fitting large designs is dramatically increased.

FITTER

The FITTER examines the architecture of the specified device, then tries to map the minimized equations into the resources available. The Fitter automatically assigns pins unless pin assignments are already specified in the design input file. The fitting sequence continues until a successful fit is accomplished or all possible implementations are exhausted. iPLS II includes a new, faster Fitter that supports the all current devices. Also included in this new Fitter is the capability to allocate p-terms to adjacent macrocells for devices such as the 5AC312 and 5AC324 that support p-term allocation.

OUTPUT FILES

— JEDEC Design File
A properly designed circuit results in the desired file from the LOC—the JEDEC Design File. The JEDEC Design File is a device-tailored EPROM cell programming map expressed in JEDEC standard format.

— Resource Utilization Report
The Resource Utilization Report gives an in-depth view of what was used inside the EPLD. Items such as device pinout, macrocell usage, and feedback arrangements are all listed. Unused resources are also listed to aid the user in adding logic or merging EPLD designs.

— Logic Equation File
The LEF file lists the logic equations after they have passed through the minimizer. It is these equations that are actually implemented in the final design.

— Compiler Error File
If a logic circuit is incorrectly designed, messages are produced by the LOC denoting the errors. To assist the redesign, these errors are placed into the Compiler Error File for later reference.

FILE MERGING

Once a design is successfully implemented, the LOC can merge it with other designs by simultaneously running the two ADF's. In this manner, LSI circuits can be broken into manageable chunks that can be implemented and tested individually. After each portion is completed, the subcircuits can be merged into one ADF to implement the total design.

III. Device Programming

After the design has been successfully entered, minimized and fitted, the designer programs his part using the JEDEC file produced by the LOC. Programming is accomplished by running the Advanced Programming Tool (APT) or Logic Programmer Software (LPS).

ADVANCED PROGRAMMING TOOL

APT is Intel's Advanced Programming Tool for programming EPLDs from JEDEC files. APT offers a command line interface with easy command syntax. It provides the ability to program, read, and verify

EPLDs via an iUP-PC Personal Programmer or iUP-200A/201A programmer using GUPI Adaptors for EPLDs. APT supports all current Intel EPLDs and new devices as they become available.

LOGIC PROGRAMMER SOFTWARE

To program a device with the LPS (available for DOS only), the user enters the file name and device to be programmed. The LPS checks if the device is blank, programs the device, then verifies that the device was programmed correctly. As a part of the Intel EPLD Programming Algorithm, each programmed cell is checked. Adding the complete device check after programming gives double verification that the part has been successfully programmed.

It is also possible to read a pre-programmed device and program other devices with the program read. The JEDEC Editor in LPS provides a hierarchical view of the device from the pin level, to the macrocell level, to the product term level. At the product term level, individual EPROM cells may be set or reset to connect or disconnect the logic equation inputs.

If the user does not want an EPLD to be read, the Security bit may be set when running the LPS. The Security Bit prevents a device from being examined after it has been programmed. This function is useful for protecting confidential designs.

iUP-PC HARDWARE

The Intel Universal Programmer for the Personal Computer consists of the PCPP programming card, 50-lead interconnect cable, GUPI base and GUPI adapter. Together they form a system for programming most PROM-type Intel devices directly from the PC host.

PCPP

The Personal Computer Personal Programmer (PCPP) is the programmer interface card that fits into the IBM AT/XT or true compatible. It is capable of driving both the iUP-GUPI base and the iUP-FAST27K personality module. The PCPP emulates the performance of the Intel iUP-200A. APT/LPS or iPPS (Intel PROM Programmer Software) controls the PCPP, causing the programming card to generate the control signals for the GUPI base.

GUPI BASE

The Generic Universal Programmer Interface (GUPI) is used for all programmable logic support. As all signal generation to devices is done by the GUPI, the programming waveforms are extremely reliable. Using the GUPI also allows upgrading for future devices with the simple addition of a plug-in adaptor. Future Intel EPLDs will be supported by the GUPI system.

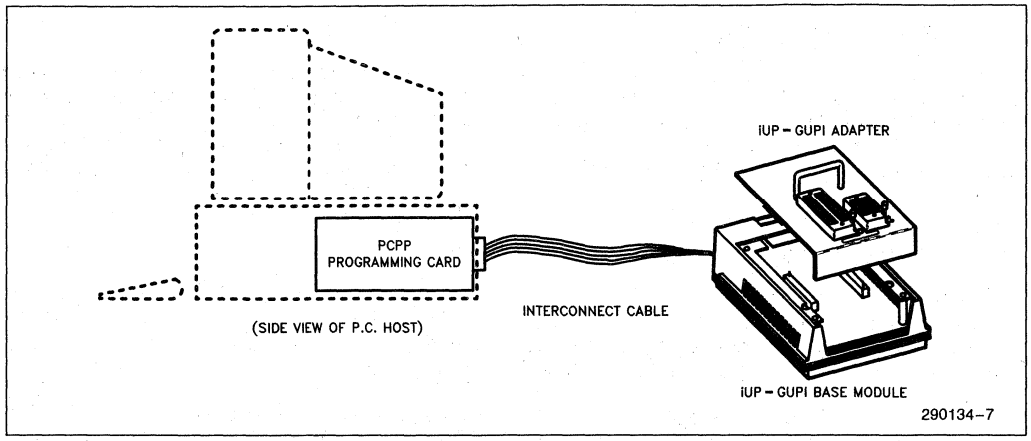
GUPI ADAPTERS

Table 1 details the GUPI adapters required for the logic devices. The adapters available for programming EPROM's and microcontrollers can be found in the data sheet for the iUP-PC (Intel order number 290130). The adapters contain the device description data for a family of similar devices.

SIMULATION

iPLS II provides a functional simulation utility (SIM) to help verify the basic operation of compiled designs. SIM uses the reduced LEF (Logic Equation File) output by the LOC, together with a user-created input vector file, to functionally simulate designs.

Full A.C. timing simulation is not supported by the SIM utility, but is available via optional products such as IPLDview-286.



The Intel Universal Programmer for the Personal Computer (iUP-PC)

TTL MACRO LIBRARY

A TTL Macro Library provides macro definitions for most common 74-series TTL devices. The library is accessed by the Macro Resolver when compiling an ADF. Macro definitions implement the TTL functions via EPLD devign primitives and Boolean equations. A complete listing of the contents of the TTL macros is provided in Applications Brief, AB-18, *TTL Macro Library Listing for EPLD Designs*.

EPLD CUSTOM MACRO LIBRARY

An EPLD Custom Macro Library provides macro definitions for a set of common EPLD primitive groups and "generic" logic functions. These functions include groups of logic gates, inputs, I/O pins, counters, multiplexers, etc. The library is accessed by the Macro Resolver when compiling an ADF. A complete listing of the EPLD custom macros is provided in Applications Brief, AB-21, *EPLD Custom Macro Library Listing for EPLD Designs*.

TEXT-BASED JEDEC EDITING

JED2AJD and AJD2JED are two utilities that support text-based editing of JEDEC files. JED2AJD translates JEDEC standard files into annotated JEDEC files, where editing can be performed via a user's standard text editor. When editing is complete, AJD2JED reconstructs the standard JEDEC file from the annotated JEDEC file, including the edits.

Table 1. Intel Programmable Logic Development System II Programming Support

Device	Number of Macrocells	iUP-GUPI Adapter	Package Type Supported
5C031, EP310	8	GUPI 20D20J	20 Pin DIP
5C032, EP320	8	GUPI 20D20J	20 Pin DIP
85C220	8	GUPI 20D20J	20-Pin DIP and PLCC
85C224	8	GUPI 24D28J	24-Pin DIP and 28-Pin PLCC
85C060	16	GUPI LOGIC-IID	24-Pin DIP
85C090	24	GUPI LOGIC-IID	40-Pin DIP
5C060, EP600	16	GUPI LOGIC-IID	24 Pin DIP
5C090, EP900	24	GUPI LOGIC-IID	40 Pin DIP
5C180, EP1800	48	GUPI LOGIC-18	68 Pin PLCC
5C180PGA	48	GUPI LOGIC-18PGA	68 Pin PGA
5AC312	12	GUPI LOGIC-IID	24 Pin DIP
5AC324	24	GUPI 40D44J	40-Pin DIP
85C508	8	GUPI 85EPLD28	28-Pin DIP and PLCC
85C960	ASPLD	GUPI 85EPLD28	28-Pin DIP and PLCC

(EPXXX Devices from Altera Corp.)

SPECIFICATIONS

Host System

Host system requirements vary, depending on the platform being used. Table 2 lists host system requirements for the platforms supported.

Table 2. iPLS/II Host System Requirements

Platform	Requirements
PC Running DOS	DOS V3.0 (or Later) 512K RAM (640K Recommended) (Color Graphics Card/Monitor Recommended) Adapter card for PC-Based Programmer (Included with iPLDS II, Optional with iPLS II) Requires One Adapter Slot.
SUN3 Workstation	SUN OS V4.0 (or Later) 4 Mbytes of RAM (8-Mbytes Recommended) iUP-200A/201A Programmer (Optional) Requires One 25-Pin Male Serial Connector for Interface.
SUN4 Workstation	SUN OS V4.0 (or Later) 4-Mbytes of RAM (8-Mbytes Recommended) iUP-200A/201A Programmer (Optional) Requires One 25-Pin Male Serial Connector for Interface.

4

Product Contents

Contents of the products vary slightly depending on the platform being used. Table 3 summarizes product contents by platform.

Table 3. iPLDS II/iPLS II Contents by Platform

Platform	IPLS	LOC	APT	LPS	MLIB	SIM	TTL	EMAC	AJED	IUPPC
iPLDS II—DOS, V2.2	X	X	X	X	X	X	X	X	X	X
iPLS II—DOS, V2.2	X	X	X	X	X	X	X	X	X	
iPLS II—SUN3, V3.0		X	X		X	X	X	X	X	
iPLS II—SUN4, V3.0		X	X		X	X	X	X	X	

IPLS = iPLS Main Menu Shell
LOC = Logic Optimizing Compiler
LPS = Logic Programming Software
APT = Advanced Programming Tool
MLIB = Macro Librarian
SIM = LEF-Based Functional Simulator
TTL = TTL Macro Library
EMAC = EPLD Custom Macro Library
AJED = JED2AJD and AJD2JED Utilities
IUPPC = PC-Based Personal Programmer and GUPI Base

Operating Environment

Electrical Characteristics

PCPP: Worst Case Power Consumption at IBM PC I/O Channel

Supply Voltage	Voltage Variance	Max. Current Drain
+5V	+5%, −4%	1.898 A
−12V	+10%, −9%	102.9 mA
+12V	+5%, −4%	530 mA

Physical Characteristics

PCPP:

Length: 13.3 inches (33.9 cm)
Height: 3.9 inches (10.0 cm)

INTERCONNECT CABLE:

50 lead ribbon cable
Length: 3.0 feet (91.4 cm)
Width: 2.43 inches (5.5 cm)

GUPI:

Length: 7.0 inches (17.8 cm)
Width: 5.5 inches (1.4 cm)
Height: 1.6 inches (4.1 cm)

Environmental Characteristics

Reading Temperature: +10°C to +40°C
Programming Temperature: +25°C ±5°C
Operating Humidity: 10% to 80% relative humidity

ORDERING INFORMATION

Order Code	Product Description
iPLDS II	Intel Programmable Logic Development System II for PCs running DOS. See Table 3 for contents.
iPLS II	Intel Programmable Logic Software (iPLS II) for PCs running DOS. See Table 3 for contents.
iPLSSUN3	Intel Programmable Logic Software (iPLS II) for the SUN3 workstation. See Table 3 for contents.
iPLSSUN4	Intel Programmable Logic Software (iPLS II) for the SUN4 workstation. See Table 3 for contents.
iUP-PC	Intel Universal Programmer for the Personal Computer: PCPP programming card, interconnect cable, iUP-GUPI base, Intel PROM Programming Software, PCPP User's Guide.
iSTATE	Intel State Machine Software: Entry format documentation, state machine convertor for LOC.
iUP-GUPI	Intel Universal Programmer—Generic Universal Programmer Interface: Generic programmer base which holds GUPI adaptors.
GUPI LOGIC-IID	GUPI Adaptor for the 5AC312, 5C060, 5C090, 85C060, and 85C090.
GUPI LOGIC-18	GUPI Adaptor for the 5C180 and future 68 pin PLCC and JLCC EPLDs.
GUPI LOGIC-18PGA	GUPI Adaptor for the 5C180 device in a 68 pin PGA package.
GUPI 20D20J	GUPI Adaptor for the 85C220; includes 20-pin DIP and JLCC sockets. Supports 5C032 and 5C031 too.
GUPI 24D28J	GUPI Adaptor for the 85C224; includes 24-pin DIP and 28-pin PLCC sockets.
GUPI 40D44J	GUPI Adaptor for the 5AC324; includes 40-pin DIP and 44-pin JLCC sockets.
GUPI 85EPLD28	GUPI Adaptor for the 85C508; includes 28-pin DIP and JLCC sockets.
ADAPT24TO28	Adapts 24 pin DIP socket to 28 pin PLCC socket; for use with GUPI LOGIC-IID.
ADAPT40TO44	Adapts 40 pin DIP socket to 44 pin PLCC socket; for use with LOGIC-IID.

iSTATE
STATE MACHINE CONVERTER SOFTWARE

- **Use State-Machine Design Techniques for Intel EPLDs**
- **Truth Table Support Including Don't Care Conditions Aid in Random Logic Design**
- **Multiple Conditional/Unconditional Branching on State Transitions**
- **Multiple State Machine, Network, and Equation Sections Supported in Same File**
- **Flexible Clock Control Allows Independent Synchronous or Asynchronous Clock for Each State Machine**
- **Independent Presets, Clears, and Output Enables Available for Each State Machine**
- **User-Defined State Assignment of State Machine Variables**
- **File Can Contain ADF Syntax Including Macros**

iSTATE is a software package that allows designers to apply state machine design methods to Intel EPLDs (Erasable Programmable Logic Devices). With iSTATE, designers create a State Machine File with an ASCII text editor and invoke iSTATE to convert the file into an ADF (Advanced Design File) for use by Intel's Logic Optimizing Compiler (LOC). The ADF is processed by the LOC to produce a JEDEC programming file and other files that document the design. Figure 1 shows EPLD design flow with iSTATE. iSTATE can be run as a standalone preprocessor for the LOC or can run under a design environment interface such as iPLS II (Intel's Programmable Logic Software II).

4

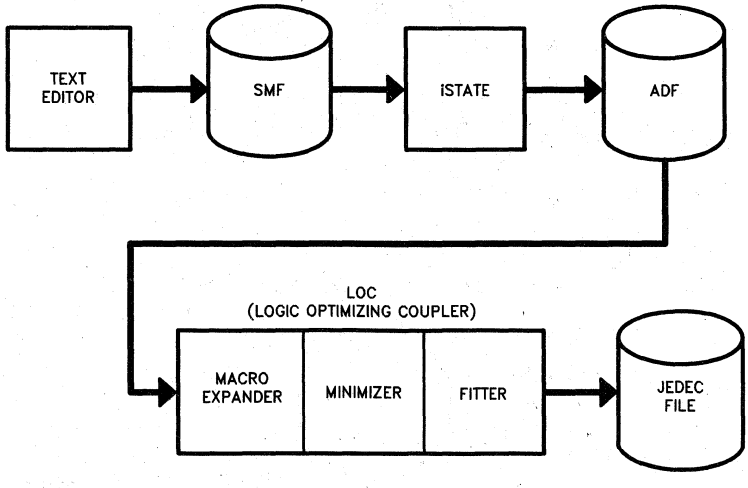


Figure 1. State Machine Design Flow Using iSTATE

August 1990
Order Number: 290249-002

State Machines

State machine design allows complex sequential operations to be expressed in clear flowcharts and diagrams. iSTATE software allows these same state machines to be expressed in a simple, straightforward syntax that can be automatically translated into ADF Network entries and Boolean logic equations and compiled by the LOC. iSTATE shortens design time and increases the reliability of state machine designs by automating the tedious step of translating state diagrams into Boolean equations.

Figure 2 shows a simple 2-bit up/down counter state diagram. Figure 3 shows how this design might be expressed in SMF (State Machine File) format for translation by iSTATE.

Header information is used to document important design information. The header is followed by declarations to define the target EPLD, input and output signals, and the state of the Turbo Bit. In the machine section, which follows, the clock and control inputs are first defined. These state machine control inputs can be independent of any other state machines in the file and can be driven by input pins, Boolean equations, or the outputs of other state machines.

Unique machine states are defined, together with output conditions for each state. This is followed by the state transition section, which contains conditional and unconditional transition statements. Conditional transitions can use an IF-THEN or CASE-ENDCASE syntax. Output signals/pins can be explicitly asserted, conditionally or unconditionally.

Truth Tables

Truth tables allow random logic to be expressed in an easily understandable form. iSTATE can translate truth tables in SMFs into Boolean equations. Figure 4 shows a simple decode table. Figure 5 shows the Boolean equations for the table produced by iSTATE. iSTATE truth tables allow don't care conditions in addition to 1s and 0s.

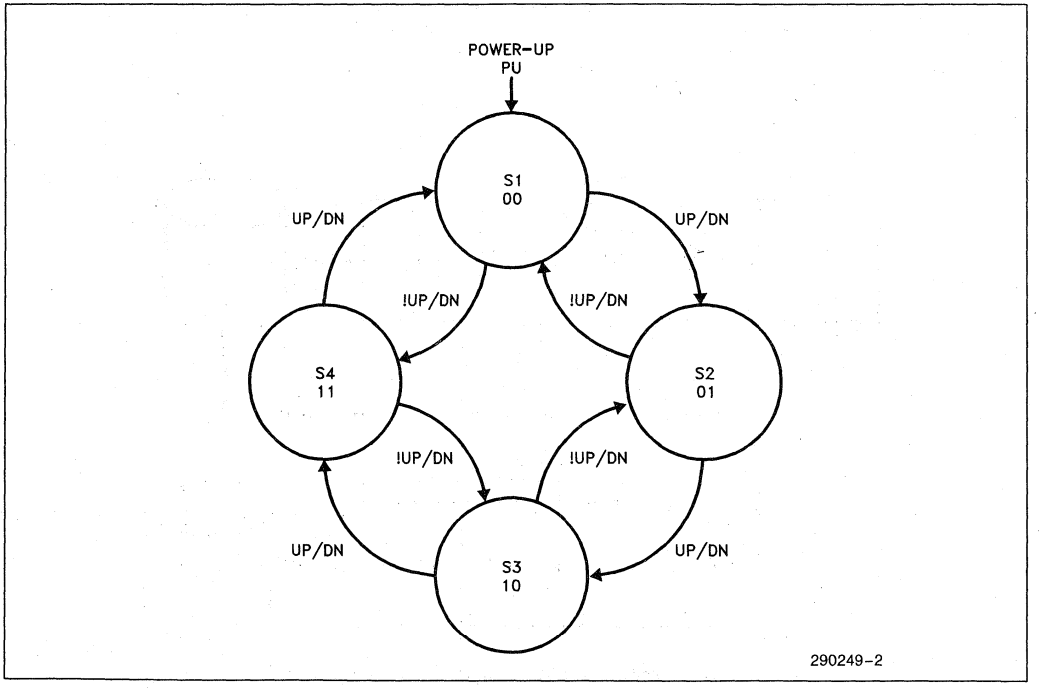


Figure 2. State Diagram for Up-Down Counter

```
YOUR NAME
YOUR COMPANY
DATE
1
A
5AC312
UDCOUNT:  2-Bit Up/Down Counter

OPTIONS:  TURBO = ON
PART:  5AC312
INPUTS:  CLK       % clock %
         UP_DOWN   % up = high; down = low %
         CLEAR     % clear %
OUTPUTS: ROLLUP    % rollover %

MACHINE:  UDCOUNT
    CLOCK:   CLK
    CLEAR:   CLEAR
    STATES:  [   Q1     Q0    ]
    S1:      [   0      0     ]
    S2:      [   0      1     ]
    S3:      [   1      0     ]
    S4:      [   1      1     ]
    S1:
        IF UP_DOWN THEN S2
        IF !UP_DOWN THEN S4
    S2:
        IF UP_DOWN THEN S3
        IF !UP_DOWN THEN S1
    S3:
        IF UP_DOWN THEN S4
        IF !UP_DOWN THEN S2
    S4:
        IF UP_DOWN THEN S1
        IF !UP_DOWN THEN S3
            ASSERT:  IF UP_DOWN THEN ROLLUP
    END$
```

Figure 3. State Machine File for Up-Down Counter

```
T_TAB:  IN1  IN2  IN3  IN4  IN5  IN6  :  OUT1  OUT2  OUT3  OUT4  ;
% row %  0    1    0    0    1    0   :   0     0     0     1    ;
         0    0    X    0    X    1   :   0     0     0     1    ;
         1    1    1    0    0    0   :   0     0     1     0    ;
         1    0    1    0    0    0   :   0     1     0     0    ;
         0    0    0    1    0    1   :   1     0     0     0    ;
         1    0    1    1    1    1   :   0     1     0     1    ;
         0    X    0    1    0    X   :   1     0     0     1    ;
```

Figure 4. Example Truth Table for Decode Logic

```
%
Boolean Equation "Truth Table"
%

OUT1 = (IN1' * IN3' * IN4 * IN5')
     + (IN1' * IN2' * IN3' * IN4 * IN5' * IN6);
OUT2 = (IN1 * IN2' * IN3 * IN4 * IN5 * IN6)
     + (IN1 * IN2' * IN3 * IN4' * IN5' * IN6');
OUT3 = (IN1 * IN2 * IN3 * IN4' * IN5' * IN6');
OUT4 = (IN1' * IN3' * IN4 * IN5')
     + (IN1 * IN2' * IN3 * IN4 * IN5 * IN6)
     + (IN1' * IN2' * IN4' * IN6)
     + (IN1' * IN2 * IN3' * IN4' * IN5 * IN6');
```

Figure 5. Boolean Equations Generated from Example Truth Table

Additional Features

iSTATE includes the following additional features:

- Macro support
- Automatic toggle flip-flop selection

iSTATE recognizes and passes through ADF macro syntax. This feature allows designers to interconnect their user-defined state machines to familiar TTL macros or user-defined macros. Use of macros simplifies design by allowing the design to be partitioned into sections that can be implemented in the most convenient manner.

During translation, iSTATE automatically selects toggle flip-flops when possible to reduce the number of p-terms required to implement the design in the target EPLD.

Order Code	Description
iSTATE	For DOS platforms (iSTATE is automatically included in iPLS II for other platforms)

iUP-PC
UNIVERSAL PROGRAMMER FOR THE PERSONAL COMPUTER

- **Personal Computer Version of the iUP-200A/201A Universal Programmers**
- **Runs on an IBM PC AT*, PC XT* or True Compatible**
- **GUPI Adaptors and Personality Modules Provide Support for Numerous Device Families**
- **Utilizes the int$_e$ligent™ and Quick-Pulse Programming™ Algorithms**
- **Extremely Versatile—Programs Intel EPROMs, EPLDs, Peripherals, and Micro-Controllers**

The Intel Universal Programmer for the Personal Computer, iUP-PC, provides a high performance programming solution from a PC host. Through plug-in adaptors for the Generic Universal Programmer Interface (iUP-GUPI), the iUP-PC supports all Intel EPLDs and most other Intel programmable devices.

290130-1

This product is manufactured by Intel Puerto Rico, Inc.

NOTE:
GUPI Adaptor NOT included.

*IBM PC/AT and PC/XT are registered trademarks of International Business Machines Corporation.

September 1990
Order Number: 290130-005

FUNCTIONAL DESCRIPTION

The iUP-PC provides a fast, versatile and reliable programming solution from a Personal Computer host. Downloading to a stand-alone programmer or moving from one workstation to another is no longer required. With the iUP-PC, the designer may do his development and programming on one workstation. Through the Generic Universal Programmer Interface (iUP-GUPI), the iUP-PC is made extremely versatile. With the iUP-GUPI the designer may program supported EPROM, Microcontroller, Peripheral and EPLDs with the mere change of a plug in adaptor. No other hardware or software addition is needed. As all of the programming signals are generated at the GUPI base, extremely reliable waveforms reach the device.

COMPONENTS

The iUP-PC programming system consists of five components:

PCPP—The Personal Computer Personal Programmer (PCPP) is an IBM PC/XT form factor expansion card which fits into an IBM PC/XT, PC/AT or true compatible.

Interconnect Cable—A 50 lead ribbon cable connects the PCPP to the iUP-GUPI.

iUP-GUPI—The Intel Universal Programmer—Generic Universal Programmer Interface (iUP-GUPI) is the programming base which holds the device adaptors.

GUPI Adaptors*—The GUPI Adaptors plug-in to the iUP-GUPI base. They carry the sockets and hardware for a particular device family.

iPPS—The Intel PROM Programmer Software (iPPS) runs on a personal computer under DOS and controls the PCPP/host communication.

***NOTE:**
Though the iUP-GUPI base is included in the iUP-PC package, the GUPI Adaptors are NOT included. The desired adaptors must be ordered separately.

PCPP CARD

The PCPP is a co-processor board. Communication between the host and the PCPP may be controlled by iPPS, LPS (Logic Programmer Software), or APT (Advanced Programming Tool). Version 2.3 or greater of iPPS is required for running the iUP-PC on a personal computer. LPS/APT are the programming software products included in Intel's Programmable Logic Software II (iPLS II). The PCPP is capable of driving the iUP-GUPI and FAST27/K modules.

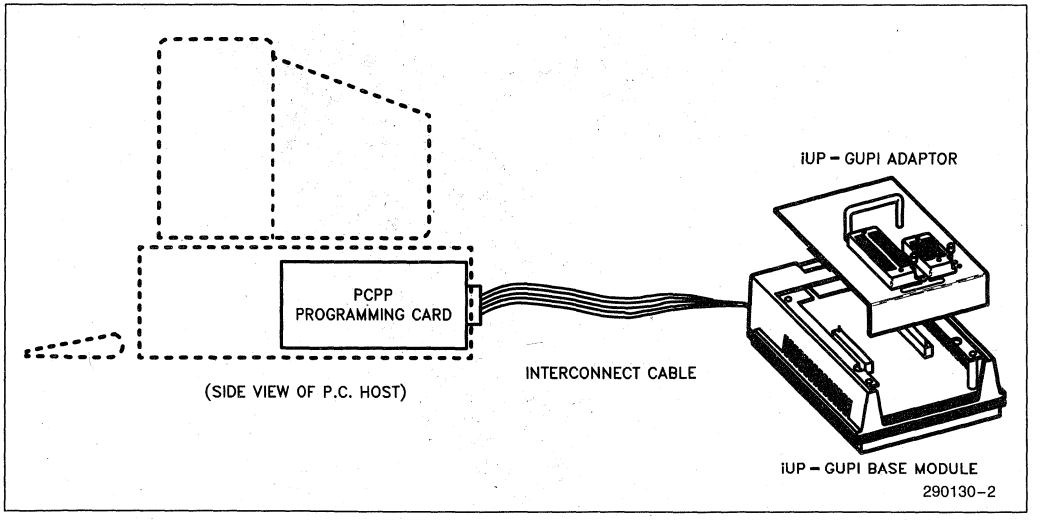


Figure 1. The Intel Universal Programmer for the Personal Computer (iUP-PC)

iUP-GUPI MODULE

The iUP-GUPI is a generic base module that enables the iUP-PC system to accept low-cost plug-in adaptors. These adaptors configure the system to support a wide variety of programmable devices —EPROMs, microcontrollers, and EPLDs—as well as device package types.

The iUP-GUPI module connects to the PCPP card via a ribbon cable. An opening in the top of the iUP-GUPI provides easy plug-in installation of the GUPI adaptors (refer to Figure 2).

The iUP-GUPI offers the programming performance of earlier Intel personality modules, with the fastest Intel programming algorithms for each device type. For example, the iUP-GUPI uses the new Quick-Pulse Programming algorithm to program the 1-Meg EPROM in seconds.

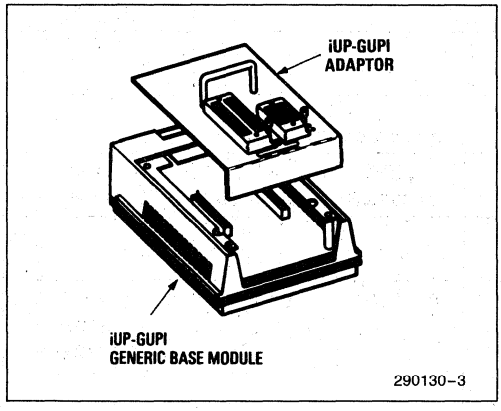


Figure 2. GUPI Adaptor Installation

GUPI ADAPTORS

The iUP-GUPI adaptors provide the final link of the iUP-PC programming system. The adaptors provide the proper sockets and characteristic information for families of Intel devices.

The iUP-GUPI LOGIC adaptors complete the programming solution of the Intel Programmable Logic Development System II (iPLDS II). The GUPI LOGIC adaptors provide support for the entire range of EPLDs. The adaptors support families of EPLDs with similar architecture, such as the 5C060 and 5C090.

Intel's one megabit EPROMs are also supported with GUPI adaptors. Adaptors are available for the 1-, 2-, and 4-Megabit EPROMs. The page mode of the 27011 is supported by the GUPI 27011 adaptor. Other Intel EPROM support is provided by the GUPI EPROM28 Adaptor.

The MCS-51 and MCS-96 microcontroller families are supported by the GUPI MSC-51 and GUPI 8796 adaptors and their derivatives. Supplemental adaptors provide support for the variety of microcontroller package types. The 8741 and 8742 peripheral components are supported by the GUPI 8742 adaptor.

Table 1 displays a cross-reference of the EPLD GUPI adaptors and the devices they support. Table 2 displays a cross-reference of the EPROM adaptors and the devices they support. Table 3 shows programming support for Microcontrollers. Note that these tables are current at the time of printing. Contact your Intel sales representative for information on current support.

4

Table 1. Programming Adaptors for EPLDs

Device Type	GUPI Logic-IID	GUPI 20D20J	GUPI 24D28J	GUPI 40D44J	GUPI Logic-18	GUPI Logic-18PGA	GUPI 85EPLD28
EPLD	5C060 85C060 5C090 85C090 5AC312	85C220	85C224	5AC324	5C180	5C180G	85C508 85C960
Package Types	DIP*	DIP PLCC	DIP PLCC	DIP PLCC	PLCC	PGA	DIP PLCC

*ADAPT units available to adapt DIP socket for PLCC package.

Table 2. Programming Adaptors for EPROMs

Device Type	GUPI 27010	GUPI 27011	GUPI 27210	GUPI 27960	GUPI EPROM28
EPROM	27010 27C010 27C100 27C020 27C040	27011	27210 27C202 27C210 27C220 27C240 87C75PF	27960CX 27960KX	2764A 27C64 87C64 27128A 27C128 27256 27C256 68C257 87C257 27512 27C512 27513 27C513 27011 27C011 27C021
Package Types	DIP	DIP	DIP		

Table 3. Programming Adaptors for Microcontrollers

Device Type	GUPI 8742	GUPI MCS-51	GUPI 8796	GUPI 87C51GB	GUPI 87C51GBP	GUPI MCS-96LCC
Peripheral	8741AH 8742AH					
Microcontroller		8751H 8751BH 87C51 8752BH 87C51FA 87C51FB	8796BH 8794BH 8795BH 8797BH	87C51GB	87C51GB	8796JC 8797BH 87C196KB
Package Types	DIP	PLCC DIP	PGA DIP	LCC	PLCC	LCC

iUP-Fast 27/K Personality Module

Devices supported in the past by the iUP-FAST 27/K Personality Module are now supported by the GUPI EPROM28 adaptor and the GUPI base module.

iPPS SOFTWARE

The iPPS software, included with the iUP-PC brings increased flexibility to PROM programming. The iPPS software provides user control through an easy-to-use interactive interface and performs the following functions to make programming quick and easy:

- Reads PROMs, ROMs and EPLDs.
- Programs PROMs directly or from a file.
- Verifies PROM data with buffer data.
- Prints PROM buffer, or device file contents on the system printer.
- Performs interactive formatting operations such as interleaving, nibble swapping, bit reversal, and block moves.
- Programs multiple PROMs from the source file, prompting the user to insert new PROMs.
- Uses a buffer to change PROM contents.

With the iPPS software the user can load programs from system memory or directly from a disk file. Access to the disk lets the user create and manipulate data in a virtual buffer. This block of data can be formatted into different PROM word sizes for program storage into several different PROM types. In addition, a program stored in the target PROM, the system memory, or a system disk file can be interleaved with a second program and entered into a specific target PROM or PROMs.

The iPPS software supports data manipulation in the following Intel formats: 8080 hexadecimal ASCII, 8080 absolute object, 8086 hexadecimal ASCII, 8086 absolute object, 80286 absolute object, and 80386 bootloadable object. Addresses and data can be displayed in binary, octal, decimal, or hexadecimal. The user can easily change default data formats as well as number bases.

iUP-PC SPECIFICATIONS

HOST SYSTEM

The iPPS will run on an IBM PC/XT, PC/AT or other true compatible with a DOS operating system. The PCPP requires one full-sized card slot inside the PC.

OPERATING ENVIRONMENT

Electrical Characteristics

PCPP:

Worst Case Power Consumption at IBM PC I/O Channel

Supply Voltage	Voltage Variance	Max. Current Drain
+5V	+5%, −4%	1.898 A
−12V	+10%, −9%	102.9 mA
+12V	+5%, −4%	530 mA

Physical Characteristics

PCPP:

Length: 13.3 inches (33.9 cm)
Height: 3.9 inches (10.0 cm)

Interconnect Cable:

50 lead ribbon cable
Length: 3.0 feet (91.4 cm)
Width: 2.43 inches (5.5 cm)

iUP-GUPI:

Length: 7.0 inches (17.8 cm)
Width: 5.5 inches (1.4 cm)
Height: 1.6 inches (4.1 cm)

Environmental Characteristics

Environmental Class: B

Temperature:

Reading	10°C to 40°C
Programming	25°C ±5°C
Operating	10°C to 40°C
Non-Operating	−40°C to 70°C

Relative Humidity:

Operating	85% Maximum
Non-Operating	95% Maximum

DOCUMENTATION

168161—PCPP User's Guide

166428—iUP-GUPI Module User's Guide

ORDERING INFORMATION

Order Code	Product Description
iUPPC	Universal Programmer for the Personal Computer: PCPP Programming Card, 50-Lead Interconnect Cable, iUP-GUPI, iPPS, PCPP User's Guide
ADAPT24TO28	28-Pin PLCC Socket Adaptor for GUPI LOGIC-IID
ADAPT40TO44	44-Pin PLCC Socket Adaptor for GUPI LOGIC-IID
piUPGUPI	Generic Universal Programmer Interface (Base)
GUPI LOGICIID	GUPI Logic Adaptor
GUPI40D44J	GUPI Logic Adaptor
GUPI85EPLD28	GUPI Logic Adaptor
GUPI 20D20J	GUPI Logic Adaptor
GUPI 24D28J	GUPI Logic Adaptor
GUPI LOGIC18	GUPI Logic Adaptor
GUPI LOGIC18PGA	GUPI Logic Adaptor for 5C180 PGA
GUPI27010	iUP-GUPI EPROM Adaptor
GUPI27011	iUP-GUPI EPROM Adaptor
GUPI27210	iUP-GUPI EPROM Adaptor
GUPI27960	iUP-GUPI EPROM Adaptor
GUPI EPROM28	iUP-GUPI EPROM Adaptor
GUPI8742	iUP-GUPI Peripheral Adaptor
GUPIMCS51	iUP-GUPI Microcontroller Adaptor
GUPI87C51GB	iUP-GUPI Microcontroller Adaptor
GUPI87C51GBP	iUP-GUPI Microcontroller Adaptor
GUPI8796	iUP-GUPI Microcontroller Adaptor
GUPI MCS-96LCC	iUP-GUPI Microcontroller Adaptor
piUPFAST 27K	(Order GUPI EPROM28 and piUPGUPI)

iUP-200A/iUP-201A UNIVERSAL PROM PROGRAMMERS

MAJOR iUP-200A/iUP-201A FEATURES:

- **Provides Programming Support for Intel and Intel-Compatible EPROMs, PLDs, Microcontrollers, and Peripherals**
- **PROM Programming Software (iPPS) Makes Programming Easy with IBM PC XT*, PC AT*, and PC Compatibles**
- **Supports Personality Modules and GUPI Base W/Adaptors**
- **iUP-200A Provides On-Line Operation with a Built-In Serial RS232 Interface and Software for a PC Environment**
- **iUP-201A Provides Same On-Line Performance and Adds Keyboard and Display for Stand-Alone Use**
- **iUP-201A Stand-Alone Capability Includes Device Previewing, Editing, Duplication, and Download from any Source Over RS232C Port**

The iUP-200A and iUP-201A universal programmers program and verify data in Intel and Intel compatible, programmable devices. The iUP-200A and iUP-201A universal programmers provide on-line programming and verification in a growing variety of development environments using the Intel PROM programming software (iPPS). In addition, the iUP-201A universal programmer supports off-line, stand-alone program editing, duplication, and memory locking. The iUP-200A universal programmer is expandable to an iUP-201A model.

210319-1

These products manufactured by Intel Puerto Rico, Inc.

*IBM PC XT and PC AT are registered trademarks of International Business Machines Corporation.

September 1990
Order Number: 210319-007

FUNCTIONAL DESCRIPTION

The iUP-200A universal programmer operates in on-line mode. The iUP-201A universal programmer operates in both on-line and off-line mode.

On-Line System Hardware

The iUP-200A and iUP-201A universal programmers are free-standing units that, when connected to a host computer with at least 64K bytes of memory, provide on-line programming and verification of Intel programmable devices. In addition, the universal programmer can read the contents of the ROM versions of supported devices.

The universal programmer communicates with the host through a standard RS232C serial data link. Different versions of the iUP-200A and iUP-201A are equipped with different cables, including the cable most commonly used for interfacing to that host. Care should be taken that the version with the correct cable for your particular system is selected, as cable requirements can vary with your host configuration.

Each universal programmer contains the CPU, selectable power supply, static RAM, programmable timer, interface for personality modules, RS232C interface for the host system, and control firmware in EPROM. The iUP-201A also has a keyboard and display.

A personality module or GUPI Adaptor adapts the universal programmer to a family of devices; it contains all the hardware and software necessary to program either a family of Intel devices or a single Intel device. The user inserts the personality module into the universal programmer front panel.

On-Line System Software

The iUP-200A and iUP201A includes your choice of one copy of Intel's PROM Programming software iPPS, selected from a list of versions for different operating systems and hosts. Each version includes the software implementation designed for that host and O.S. and the RS232C cable most commonly used. Additional versions may be purchased separately if you decide to change hosts at a later date. The iPPS software provides user control through an easy-to-use interactive interface. The iPPS software performs the following functions to make EPROM programming quick and easy:

- Reads devices
- Programs devices directly or from a file
- Verifies device data with buffer data
- Locks device memory from unauthorized access (on devices which support this feature)
- Prints device contents on the network or development system printer
- Performs interactive formatting operations such as interleaving, nibble swapping, bit reversal, and block moves
- Programs multiple devices from the source file, prompting the user to insert new devices
- Uses a buffer to change device contents

All iPPS commands, as well as program address and data information, are entered through the host system ASCII keyboard and displayed on the system CRT.

The iPPS software supports data manipulation in the following Intel formats: 8080 hexadecimal ASCII, 8080 absolute object, 8086 hexadecimal ASCII, 8086 absolute object, and 80286 absolute object. Addresses and data can be displayed in binary, octal, decimal, or hexadecimal. The user can easily change default data formats as well as number bases. iPPS can also access disk files.

For programming Intel EPLDs, the iUP-200A/201A can be controlled by Intel's Logic Programming Software (LPS) and Advanced Programming Tool (APT). LPS and APT program EPLDs from JEDEC files produced by Intel's logic compiler. (iPPS can also program EPLDs, but only from pre-programmed device masters.)

System Expansion

The iUP-200A universal programmer can be easily upgraded (by the user) to an iUP-201A universal programmer for off-line operation. The upgrade kit (iUP-PAK-A) is available from Intel or your local Intel distributor.

Off-Line System

The iUP-201A universal programmer has all the on-line features of the iUP-200A universal programmer plus off-line editing, device duplication, program verification, and locking of device memory independent of the host system. The iUP-201A universal programmer also accepts Intel hexadecimal programs developed on non-Intel development systems. Just a few keystrokes download the program into the iUP RAM for editing and loading into a device.

Off-line commands are entered via a 16-character keypad. A 24-character display shows programmer status.

PERSONALITY MODULES

For some devices, a personality module is the interface between the iUP-200A/iUP-201A universal programmer and a selected device. Personality modules contain all the hardware and firmware for reading and programming a family of Intel devices. Table 1 lists the devices supported by the different modules.

For most devices, the GUPI module and interchangeable GUPI Adaptors provide the interface between the programmer and the device being programmed (see Figure 1). the GUPI (Generic Universal Programmer Interface) module is a base module that intefaces to the iUP-200A/201A and GUPI Adaptors. GUPI Adaptors tailor the GUPI module base signals to a family of devices or an individual device. The GUPI module and GUPI Adaptors provide a lower-cost method of device support than if unique Personality Modules were offered for each device/family. Tables 2 through 4 show which Adaptors support which devices. Note that these tables are current at the time of printing. Contact your Intel sales representative for information on current support.

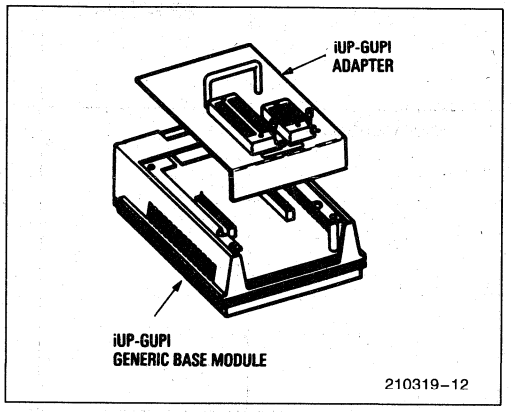


Figure 1. GUPI Adaptor

4

Table 1. iUP Personality Programming Modules

Device Type	Fast 27/K Module	F27/128 Module	F87/44A Module	F87/51A Module
EPROM	See GUPI EPROM28	2716 2732 2732A 2764 27128		
Microcontroller			8041A 8042 8044AH 8741H 8742 8744H 8755A	8748 8748H 8749H 8751 8751H 8048 8048H 8049 8049H 8050H 8051

*Quick-Pulse Programming™ algorithm

Table 2. Programming Adaptors for Memories

Device Type	GUPI 27010	GUPI 27011	GUPI 27210	GUPI 27960	GUPI EPROM28
EPROM	27010 27C010 27C100 27C020 27C040	27011	27210 27C202 27C210 27C220 27C240 87C75PF	27960CX 27960KX	2764A 27C64 87C64 27128A 27C128 27256 27C256 68C257 87C257 27512 27C512 27513 27C513 27011 27C011 27C021
Package Types	DIP	DIP	DIP		

Table 3. Programming Adaptors for EPLDs

Device Type	GUPI Logic-IID	GUPI 20D20J	GUPI 24D28J	GUPI 40D44J	GUPI Logic-18	GUPI Logic-18PGA	GUPI 85EPLD28
EPLD	5C060 85C060 5C090 85C090 5AC312	85C220	85C224	5AC324	5C180	5C180G	85C508 85C960
Package Types	DIP*	DIP PLCC	DIP PLCC	DIP PLCC	PLCC	PGA	DIP PLCC

*ADAPT units available to adapt DIP socket for PLCC package.

Table 4. Programming Adaptors for Microcontrollers

Device Type	GUPI 8742	GUPI MCS-51	GUPI 8796	GUPI 87C51GB	GUPI 87C51GBP	GUPI MCS-96LCC
Peripheral	8741AH 8742AH					
Microcontroller		8751H 8751BH 87C51 8752BH 87C51FA 87C51FB	8794BH 8795BH 8796BH 8797BH	87C51GB	87C51GB	8796JC 8797BH 87C196KB
Package Types	DIP	PLCC DIP	PGA DIP	LCC	PLCC	LCC

4

iUP-200A/iUP201A SPECIFICATIONS

Control Processor

Intel 8085A microprocessor
6.144 MHz clock rate

Memory

RAM—4.3 bytes static
ROM—12K bytes EPROM

Interfaces

Keyboard: 16-character hexadecimal and 12-function keypad (iUP-201A model only)

Display: 24-character alphanumeric (iUP-201A model only)

Software

Monitor— system controller in pre-programmed EPROM

iPPS — Intel PROM programming software on supplied diskette

Physical Characteristics

Depth: 15 inches (38.1 cm)

Width: 15 inches (38.1 cm)

Height: 6 inches (15.2 cm)

Weight: 15 pounds (6.9 kg)

Electrical Characteristics

Selectable 100, 120, 200, or 240 Vac ± 10%; 50-60 Hz

Maximum power consumption—80 watts

Environmental Characteristics

Reading Temperature:	10°C to 40°C
Programming Temperature:	25°C ±5°
Operating Humidity:	10% to 85% relative humidity

Reference Material

166041-001— *iUP-200A/201A Universal Programmer User's Guide.*

166042-001— *Getting Started with the iUP-200A/201A (For ISIS/iNDX Users).*

166043-001— *Getting Started with the iUP-200A/201A (For DOS Users).*

164853 — *iUP-200A/201A Universal Programmer Pocket Reference.*

ORDERING INFORMATION

Product Order Code	Description
iUP-200A 216D	On-Line PROM programmer with iPPS rel 2.0 for PC/DOS, and cable for PC or XT
iUP-200A 217D	On-Line PROM programmer with iPPS rel 2.0 for PC/DOS, and cable for AT
iUP-201A 216D	Off-Line and on-line PROM programmer with iPPS rel 2.0 for PC/DOS, and cable for PC or XT
iUP-201A 217D	Off-Line and on-line PROM programmer with iPPS rel 2.0 for PC/DOS, and cable for AT
iUP-200/201 U1* Upgrade Kit	Upgrades an iUP-200/201 universal programmer to an iUP-200A/201A universal programmer
iUP-DL	Download Support Kit for iUP-200A/201A upgrades programmer to support adaptors that use software programming (.DSS) files.
iUP-PAK-A Upgrade Kit	Upgrades an iUP-200/A universal programmer to an iUP-201A universal programmer

*Most personality modules can be used only with an iUP-200A/201A universal programmer or an iUP-200/iUP201 universal programmer upgraded to an A with the iUP-200/iUP-201 U1 upgrade kit.

Product Order Code	Description
piUP-GUPI	Generic Universal Programmer Interface (Base)

Software Sold Separately

Product Order Code	Description
216D	PROM programming software rel 2.0 for PC/DOS with cable for PC or PC XT
217D	PROM programming software rel 2.0 for PC/DOS with cable for PC AT

4

intel®

September 1990

Programmable Logic Tools Product Briefs

Order Number: 296460-002

PROGRAMMABLE LOGIC TOOLS PRODUCT BRIEFS

CONTENTS

4

IPLDview-286
Integrated Schematic Capture and Simulation Software for EPLDs

IPLDview-286 provides a powerful, PC-based, combined, schematic drawing and full-function simulation package for EPLD designs. IPLDdraw, the schematic capture program, supports multiple levels of hierarchy for both symbols and schematics. IPLDsim, the simulation program, and a waveform editor support both logic and timing simulation. Full integration provides a consistent user-friendly interface to open multiple schematic and simulation windows and allows the IPLDsim to back-annotate simulation values into the original schematics.

Creating EPLD designs is an easy task with the help of IPLDview-286's powerful schematic capture program IPLDdraw. IPLDdraw's editing commands include array commands, avoidance routing, snap-to-pin, pan and zoom, context window, cut-and-paste, and dialog boxes. Tasks can be performed via the mouse or the keyboard. Drawings can be rotated and scaled on-screen on standard or custom-sized pages. Menus can be customized to match the expertise of the user. Design errors are caught by the logical design rule checking software. ADFs (Advanced Design Files), for use by Intel's LOC (Logic Optimizing Compiler), are generated by IPLDdraw's netlist program.

IPLDsim, the IPLDview-286 simulation program, includes a user-friendly graphic interface and supports input/output waveform processing (via a waveform editor) for both logic and timing simulation of Intel EPLDs. Two methods of simulation are supported: pre-compilation simulation from an IPLDdraw schematic, and post-compilation simulation from a compiled JEDEC file. IPLDview-286 includes device architecture and delay models required for post-compilation simulation. A rich command set lets you control all facets of simulation, including determining timing delays, setting breakpoints, and logging simulation results and commands.

To protect your investment, all EPLD symbols, schematics, and simulation command files created under IPLDview-286 are fully-compatible with the Workview* environment by View*logic**. Drawings and waveform files created by IPLDview-286 for EPLDs can be used in PCB or system designs in the Workview environment.

System Requirements: IPLDview-286 runs in 80286 PCs (in protected-mode) and requires 2.0 Mbytes (expandable to 16 Mbytes) of *extended* RAM. DOS version 3.0 (or greater) is required to begin execution. IPLDview-286 is shipped on 5-¼" 1.2 Mbyte floppy disk drives.

Order Code: **IPLDview286** (Schematic Capture S/W, ADF Netlister, Symbol Libraries, Simulation Software, Waveform Editor, and TTL Macro/EPLD Simulation Models)

NOTE:

Upgrades to IPLDview-286 for users who have previously purchased IPLDdraw are available. Contact your local authorized Intel sales office.

*Workview and Viewlogic are registered trademarks of Viewlogic Systems Inc.

IPLDdraw
EPLD Schematic Capture Software

IPLDdraw provides a powerful, multi-window, PC-based schematic drawing environment for creating EPLD designs. It provides a consistent, user-friendly interface and supports multiple levels of hierarchy for schematics. IPLDdraw includes the TTL macro and EPLD design primitive symbol libraries needed to design with Intel EPLDs.

Creating EPLD designs is an easy task with the help of IPLDdraw's advanced editing commands, including array commands, avoidance routing, snap-to-pin, pan and zoom, context window, cut-and-paste, and dialog boxes. Tasks can be performed via the mouse or the keyboard. Drawings can be rotated and scaled on-screen on standard or custom-sized pages. Menus can be customized to match the expertise of the user.

IPLDdraw catches design errors with its logical design rule checking software and netlists to ADF (Advanced Design File) format for use by Intel's LOC (Logic Optimizing Compiler).

To protect your investment, all EPLD symbols and schematics created with IPLDdraw are fully-compatible with IPLDview-286, the integrated schematic capture and simulation package. They are also compatible with the Workview environment by View*logic*. Drawings created for EPLDs can be used in PCB or system designs in the Workview environment.

System Requirements: IPLDdraw runs under version 3.0 (or greater) of DOS and requires 640 Kbytes of RAM. IPLDdraw is shipped on 5-¼" 1.2 Mbyte floppy disks.

Order Code: **IPLDdraw** (Includes Schematic Capture S/W, ADF Netlister, and Symbol Libraries)

SCHEMA III-PLD

SCHEMA III-PLD is a low-cost schematic capture software for designing with Intel EPLDs and with standard MSI, SSI, and discrete components. For EPLD designs, SCHEMA III-PLD outputs Advanced Design Files (ADFs) that can subsequently be compiled by iPLS II software. Figure 1 shows the flow to generate a drawing file and convert it to an ADF for processing by iPLS II. SCHEMA III-PLD supports EPLD design primitive symbols as well as MSI and SSI macro symbols, allowing designers to combine TTL and EPLD symbols as needed. An EPLD Custom library supports groups of EPLD symbols and "generic" function symbols such as counter, multiplexers, etc. The ability to create user-defined symbols that can be translated into ADF macro calls adds to SCHEMA III-PLD's power and versatility.

SCHEMA III-PLD provides fast, smooth panning, combined mouse/keyboard support, instant command execution, and automatic "step and repeat" to make schematic capture as quick and easy as possible. In addition to the symbol libraries targeted for EPLD design, SCHEMA III-PLD provides over 10 symbol libraries for standard PCB design. Its sophisticated library management routines, reentrant object editor, and true "hierarchical" design capability makes SCHEMA III-PLD a powerful tool for professional designers.

The EPLD Manager software included with SCHEMA III-PLD provides a single user interface to both SCHEMA III-PLD and iPLS II software modules. EPLD Manager software is also available separately to users who already own SCHEMA III.

Order Codes: SCHEMA III-PLD (SCHEMA III and EPLD Manager)

EPLDMGR (EPLD Manager)

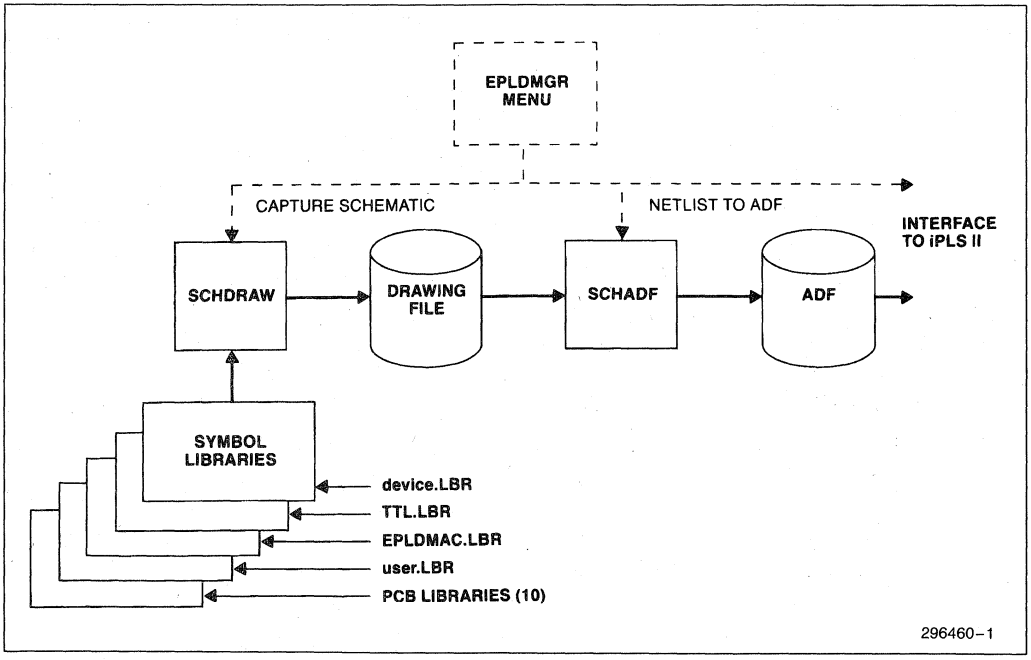


Figure 1. SCHEMA III-PLD Schematic Capture Flow for EPLD Designs

SSC1000
iPLS II Software Support Contract (MS-DOS)

SSC1000 is a software support contract for iPLS II (Intel Programming Logic Software II) on the IBM PC/AT platforms. SSC1000 guarantees that you will receive all releases and updates of our iPLS II product automatically during the term of the contract. SSC1000 covers tools and utilities bundled with iPLS II. OEM packages such as schematic capture/timing simulation software and software options such as iSTATE are not covered by SSC1000.

Order Code: **SSC1000**

APPLICATION BRIEF

AB-18

October 1988

TTL Macro Library Listing for EPLD Designs

PROGRAMMABLE LOGIC APPLICATIONS
INTEL CORPORATION

Order Number: 292037-003

TTL MACRO LIBRARY LISTING FOR EPLD DESIGNS

CONTENTS

TTL Macros

The following is a list of TTL macros that are in TTL.LIB version 3.6. This library is available through the Intel EPLD customer hot line.

These macros are called from an Advanced Design File (ADF). Schematic capture packages such as Schema II-PLD create ADFs with the correct macro invocation for each TTL device listed here.

Macros listed here are grouped by general function.

SSI GATES

7400	2 Input NAND
7402	2 Input NOR
7404	1 Input INVERTER
7408	2 Input AND
7410	3 Input NAND
7411	3 Input AND
7420	4 Input NAND
7421	4 Input AND
7427	3 Input NOR
7430	8 Input NAND
7432	2 Input OR
7486	2 Input XOR

MSI FUNCTIONS

Decoders/Demultiplexers

7442	(10) BCD to Decimal
7444	(10) Excess-3-Gray to Decimal
7447X	(7) BCD to 7-Segment—Active Low Output
7449	(7) BCD to 7-Segment—Active High Output
74138	(8) 1-of-8 Decoder
74139	(4) Single 1-of-4 Decoder
74145	(10) BCD to Decimal
74154	(16) 1-of-16 Decoder
74155	(8) Dual 1-of-4
74156	(8) Dual 1-of-4

Multiplexers

74151	(2) 8-to-1
74153	(2) Dual 4-to-1—Active High Output
74157	(4) Quad 2-to-1—Active High Output
74158	(4) Quad 2-to-1—Active Low Output
74253	(2) Dual 4-to-1—Three-State Output
74257X	(4) Quad 2-to-1—Active High, Three-State Output
74258X	(4) Quad 2-to-1—Active Low, Three-State Output
74298XA	(4) Quad 2-to-1—Active High with Storage
74298XB	(4) Quad 2-to-1—Active High with Storage
74352	(2) Dual 4-to-1—Active Low Output

Counters

		Type	Clear	Load	Clk	Extras
7490XD	(4)	BCD Decade	S	9	R	
7490XQ	(4)	Bi-Quinary	S	9	R	
74160	(5)	BCD Decade	A	S	R	RCO
74161	(5)	4-Bit Binary	A	S	R	RCO
74162	(5)	BCD Decade	S	S	R	RCO
74163	(5)	4-Bit Binary	S	S	R	RCO
74168	(5)	BCD Decade	—	S	R	U/D, RCO
74169	(5)	4-Bit Binary	—	S	R	U/D, RCO
74176XD	(4)	BCD Decade	A	S	R	
74176XQ	(4)	Bi-Quinary	A	S	R	
74177X	(4)	4-Bit Binary	A	S	R	
74190XA	(6)	BCD Decade	—	S	R	U/D, RCO, MM
74190XB	(6)	BCD Decade	—	S	R	U/D, RCO, MM
74191XA	(7)	4-Bit Binary	—	S	R	U/D, RCO, MM
74290XD	(4)	BCD Decade	S	9	R	
74290XQ	(4)	Bi-Quinary	S	9	R	
74390X	(4)	Bi-Quinary/BCD	A	—	F	
74393XA	(4)	4-Bit Binary	A	—	F	
74393XB	(4)	4-Bit Binary	A	—	F	

S = Synchronous
A = Asynchronous
9 = Synchronous Set-to-9

R = Rising-Edge Triggered
F = Falling-Edge Triggered

U/D = Up/Down
RCO = Ripple Carry Output
MM = Max/Min Output

Single Flip-Flops

7472XA	(2) AND-Gated JK Master/Slave
7472XB	(2) AND-Gated JK Master/Slave
7473X	(2) JK with Clear
7474X	(2) D with Preset and Clear
74112XA	(3) JK with Preset and Clear
74112XB	(2) JK with Clear

Latches

7475X	(8) 4-Bit Bistable
7477X	(4) Quad D-Type
74259XA	(8) Octal Addressable D-Type
74259XB	(8) Octal Addressable D-Type
74373X	(8) Octal D-Type

Multiple Flip-Flops (Registers)

74174X	(6) Hex D
74175X	(8) Quad D with Q and /Q
74273X	(8) Octal D
74377	(8) Octal D with Common Enable
74378	(6) Hex D

Shift Registers

7491	(8) 8-Bit—Serial-In, Serial-Out
7495XA	(4) 4-Bit—Serial-In/Parallel-In, Parallel-Out
7495XB	(4) 4-Bit—Serial-In/Parallel-In, Parallel-Out
7495XC	(4) 4-Bit—Serial-In/Parallel-In, Parallel-Out
7496X	(5) 5-Bit—Serial-In/Parallel-In, Parallel-Out
74164	(8) 8-Bit—Serial-In, Parallel-Out
74165X	(9) 8-Bit—Serial-In/Parallel-In, Serial-Out
74194	(4) 4-Bit Bi-Directional—Serial-In/Parallel-In, Parallel-Out
74395XA	(5) 4-Bit Cascadable—Serial-In/Parallel-In, Parallel-Out
74395XB	(5) 4-Bit Cascadable—Serial-In/Parallel-In, Parallel-Out

Miscellaneous

7482X	(4) 2-Bit Adder
7483X	(8) 4-Bit Adder
7485X	(7) 4-Bit Magnitude Comparator
7487	(4) 4-Bit True/Complement Element
74143X	(17) 4-Bit Counter; 4-Bit Latch; 7 Segment Decoder
74180X	(4) 8-Bit Parity Generator/Checker
74180XA	(4) 8-Bit Parity Generator/Checker
74182	(5) Look-Ahead Carry Generator
74183	(2) Single-Bit Full Adder with Carry/Save
74280X	(5) 9-Bit Odd/Even Parity Generator/Checker

DEMORGAN EQUIVALENTS (BUBBLE GATES)

	Bubble AND (NOR)	Bubble NAND (OR)	Bubble NOR (AND)	Bubble OR (NAND)
2 Input	BAND2	BNAND2	BNOR 2	BOR2
3 Input	BAND3	BNAND3	BNOR 3	BOR3
4 Input	BAND4	BNAND4	BNOR 4	BOR4
6 Input	BAND6	BNAND6	BNOR 6	BOR6
8 Input	BAND8	BNAND8	BNOR 8	BOR8
12 Input	BAND12	BNAND12	BNOR 12	BOR12

INPUT/OUTPUT MACROS

INPUT	N/A Generates Input Pin and Node in ADF
OUTPUT	(1) Generates Enabled Output Buffer in ADF
OUTP	(1) Output Pin (Used in SCHEMA II-PLD)
74125	(1) Single Three-State Output, Active Low Enable
74126	(1) Single Three-State Output, Active High Enable

NOTES:

1. All TTL macros duplicate TTL function only. They DO NOT DUPLICATE performance characteristics such as open-collector, totem-pole, or high-drive output.
2. Any TTL macros which deviate in some way from standard TTL function are denoted with an appended "X" (see device .DOC file for details). Appended "D"s and "Q"s indicate counters configured to Decimal or bi-Quinary mode; appended "A"s and "B"s indicate a macro configured for a family of EPLD devices (e.g. 5C060, 5C090, 5C180).
3. The (#) indicates the maximum number of EPLD macrocells consumed if all outputs are used. If an output is not used, the macro compression phase of the Macro Expander will remove the signal unless it is used as feedback inside the macro definition.
4. /Q's should be avoided as pin outputs if possible. The EPLD is structured such that the Q is readily available as a pin output and both the Q and /Q are readily available as feedbacks. Using /Q as a pin output, however, requires an extra macrocell and adds to the propagation delay.

December 1988

EPLD Custom Macro Library Listing for EPLD Designs

PROGRAMMABLE LOGIC APPLICATIONS
INTEL CORPORATION

Order Number: 292050-001

EPLD CUSTOM MACRO LIBRARY LISTING FOR ELPD DESIGNS

CONTENTS

EPLD CUSTOM MACROS

The following is a list of the macros contained in version 1.0 of Intel's EPLD Custom Macro Library (EPLDMAC.LIB). This library is available through the Intel EPLD customer hot line.

These macros are called from an Advanced Design File (ADF). Schematic capture packages such as SCHEMA II-PLD create an ADF with the correct macro invocation syntax for each macro listed here.

The macros are grouped by function. The macro name is followed by the least number of macrocells used and a description of the macro's function.

INPUTS

2INP	(0)	2 Input Pins
4INP	(0)	4 Input Pins
6INP	(0)	6 Input Pins
8INP	(0)	8 Input Pins

BURIED FEEDBACK

4NOCF	(4)	4 "No Output Combinational Feedback" I/O Primitives
6NOCF	(6)	6 "No Output Combinational Feedback" I/O Primitives
8NOCF	(8)	8 "No Output Combinational Feedback" I/O Primitives

COMBINATIONAL I/O

4CONF	(4)	4 "Combinational Output No Feedback" I/O Primitives
6CONF	(6)	6 "Combinational Output No Feedback" I/O Primitives
8CONF	(8)	8 "Combinational Output No Feedback" I/O Primitives
4COIF	(4)	4 "Combinational Output Input Feedback" I/O Primitives
6COIF	(6)	6 "Combinational Output Input Feedback" I/O Primitives
8COIF	(8)	8 "Combinational Output Input Feedback" I/O Primitives

REGISTERED I/O

4RONF	(4)	4 "Registered Output No Feedback" I/O Primitives
6RONF	(6)	6 "Registered Output No Feedback" I/O Primitives
8RONF	(8)	8 "Registered Output No Feedback" I/O Primitives
4ROIF	(4)	4 "Registered Output Input Feedback" I/O Primitives
6ROIF	(6)	6 "Registered Output Input Feedback" I/O Primitives
8ROIF	(8)	8 "Registered Output Input Feedback" I/O Primitives
4RORF	(4)	4 "Registered Output Registered Feedback" I/O Primitives
6RORF	(6)	6 "Registered Output Registered Feedback" I/O Primitives
8RORF	(8)	8 "Registered Output Registered Feedback" I/O Primitives

LATCHES/REGISTERS

4REG	(4)	4 Registers with Common Clock and Clear
6REG	(6)	6 Registers with Common Clock and Clear
8REG	(8)	8 Registers with Common Clock and Clear
4LATCH	(4)	4 Transparent Data Latches with Common Enable
6LATCH	(6)	6 Transparent Data Latches with Common Enable
8LATCH	(8)	8 Transparent Data Latches with Common Enable
8TRANS	(8)	8-Bit Bi-Directional Data Transceiver
RSLATCH	(1)	Set-Reset Latch
DLATCH	(1)	Standard D-Type, Transparent Latch
DFFPRE	(2)	D Flip-Flop with Preset and Clear

MULTIPLEXERS/ENCODERS

2MUX	(0)	2-to-1 Multiplexer
D2MUX	(0)	Two 2-to-1 Multiplexers with Common Select
Q2MUX	(0)	Four 2-to-1 Multiplexers with Common Select
4MUX	(0)	4-to-1 Multiplexer
8MUX	(0)	8-to-1 Multiplexer
16MUX	(0)	16-to-1 Multiplexer
10MUXBCD	(0)	10-to-4 BCD Encoder

CONVERTERS/DECODERS

BINGRY	(0)	4-Bit Binary to Gray Code Converter
GRYBIN	(0)	4-Bit Gray Code to Binary Converter
1DEC	(0)	1-to-2 Decoder
2DEC	(0)	2-to-4 Decoder
4DEC	(0)	4-to-16 Decoder
3DEC	(0)	3-to-8 Decoder
7SEG	(0)	4-Bits to Seven Segment Display Decoder

COUNTERS/DIVIDERS

2CNT	(2)	2-Bit Counter with Preload and Clear
4CNT	(4)	4-Bit Counter with Preload and Clear
8CNT	(8)	8-Bit Counter with Preload and Clear
16CNT	(16)	16-Bit Counter with Preload and Clear
BCDCNT	(4)	4-Bit BCD Counter with Preload and Clear
FDIV2	(4)	Divides Input Frequency By 2, 4, 8, and 16
FDIV5	(4)	Divides Input Frequency By 5, 10, 15, and 20

SHIFT REGISTERS

2SHIFT	(2)	2-Bit Serial or Parallel In Shift Register with Enable
4SHIFT	(4)	4-Bit Serial or Parallel In Shift Register with Enable
8SHIFT	(8)	8-Bit Serial or Parallel In Shift Register with Enable
16SHIFT	(16)	16-Bit Serial or Parallel In Shift Register with Enable

ARITHMETIC OPERATIONS

1ADD	(0)	1-Bit Full Adder
2MULT	(0)	2-Bit Multiplier
4COMP	(0)	4-Bit Magnitude Comparator...Equality Only
8COMP	(2)	8-Bit Magnitude Comparator...Equality Only
8PAREVN	(2)	8-Bit Even Parity Generator
8PARODD	(2)	8-Bit Odd Parity Generator

intel®

APPLICATION NOTE

AP-311

October 1988

Using Macros in EPLD Designs

DANIEL E. SMITH
PROGRAMMABLE LOGIC APPLICATIONS
INTEL CORPORATION

Order Number: 292039-002

USING MACROS IN EPLD DESIGNS

CONTENTS

4

INTRODUCTION

The iPLS II (Intel Programmable Logic Software) Logic Optimizing Compiler includes a Macro Expander that supports the use of macros in EPLD designs. This application note shows how to use the TTL and EPLD Custom macros available from Intel with ADFs created by a text editor. Included are descriptions of macro file support, guidelines for using macros, and two design examples.

OVERVIEW

iPLS II allows designers to include macro calls in design files to implement common circuit functions. Macro calls are subsequently expanded by the LOC (Logic Optimizing Compiler) into ADF network and/or equation entries required to perform the desired functions. Use of macros allows designs to proceed at a high level, which simplifies and shortens the design process. Macros can be connected together or used in conjunction with standard iPLS II EPLD primitives. Designing with macros is analogous in many ways to using subroutines in software.

Macros can be used in ADFs (Advanced Design Files) created by a text editor, or by several schematic capture software products. This application note covers use of macros in ADFs created by a text editor. Macro support at this level includes the following:

- A TTL macro library (TTL.LIB) for designing with common TTL circuit equivalents
- An EPLD custom macro library (EPLDMAC.LIB) for designing with "generic" macros.
- A Macro Expander in the LOC that expands macro calls in ADFs with the contents of the corresponding macros from libraries.

Figure 1 shows text editor/ADF macro support for iPLS II. Note that the ADF can be created by any standard ASCII text editor (text editor supplied by user). Creation of user-defined macros is covered in application note, AP-312 "Creating Macros for EPLD Designs", order number 292040. Use of macros with schematic capture software is covered in the documentation for the respective software package.

This note discusses use of macros under the following headings:

- Macro Libraries, briefly describes the two libraries available from Intel.
- Using Macros, describes macro files, how to call macros, the process of macro expansion, calling multiple macro calls, and some basic guidelines to follow and pitfalls to avoid.
- Two examples showing use of TTL macros, and mixing macros and EPLD primitives.

MACRO LIBRARIES

Intel offers two macro libraries: a TTL Library and an EPLD Custom Library.

TTL Macro Library

A TTL macro library (TTL.LIB) is available from Intel to support design entry using familiar 74-series logic

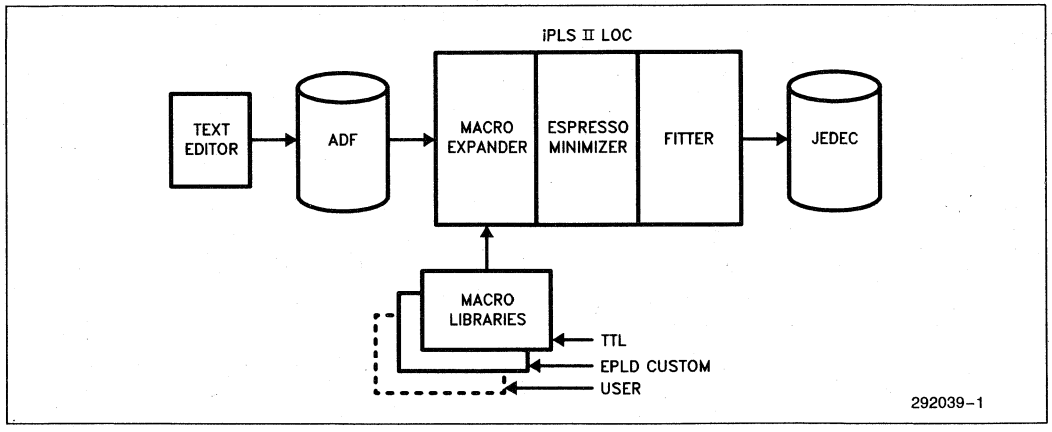


Figure 1. Text Editor/ADF Macro Support for iPLS II

devices. The library contains macros that implement the most widely used 74-series device functions as well as macros for some members of other logic families. Each device in the library is supported by a .DOC file. The .DOC file describes the macro syntax and lists any notable differences between the macro implementation and the TTL part.

EPLD Custom Macro Library

An EPLD custom macro library (EPLDMAC.LIB) is available from Intel to support design entry using groups of EPLD primitives or "generic" functions such as latches, registers, counters, decoders, etc.

USING MACROS

The iPLS II Macro Expander is automatically invoked by the LOC when an ADF is submitted to the compiler. When invoked, the Macro Expander identifies macro calls in ADFs, searches macro libraries for a corresponding macro, and expands the call with ADF network and equation entries from the macro file. The expanded file is then compiled normally.

Macro Files

Figure 2 shows the macro file for a 74138 TTL device, a commonly used one-of-eight decoder. Note that the first line contains the name and I/O signals for the device. Signals are listed in the order in which they appear on the actual TTL device, including VCC and GND (i.e., A = pin 1, B = pin 2, ..., VCC = pin 16). The sequence of signals in this line determines how the macro is "called" from an ADF.

Some of the macros in the TTL library have an "X" suffix appended to the filename, for example 74138X. This suffix indicates that the macro is device-specific (not supported on all EPLDs) or that there is some difference from the TTL device. This information is described in the .DOC file for each macro.

The second line of the macro file contains defaults for each input and place holders (blanks) for each output. The default for an input sets the input to an intelligent level (i.e., enables are enabled, clears, preset, loads are disabled, etc.).

Macro files can contain a Network section, an Equation section, or both. A Network section is not needed when the macro functions can all be implemented in Boolean equations. When used, the Network section contains EPLD design primitives. An Equations section is not needed when the macro functions can all be implemented in the Network section. Macro files end with the keyword "ENDEF".

Macro Calls

All macro calls appear in the Network section of an ADF. Macro calls use the same part/function name and signal sequence used on the first line of the macro file. The signal names in the macro and the macro call do not need to match, but the order of signals in the call is **crucial** to proper implementation of the macro function. For example, the macro call for the 74138 device could be any one of the following examples:

```
74138(A,B,C,G2A,G2B,G1,Y7,GND,Y6,Y5,
Y4,Y3,Y2,Y1,Y0,VCC)

74138(D1,D2,D3,EN1,EN2,EN3,07,GND,06,
05,04,03,02,01,00,VCC)
```

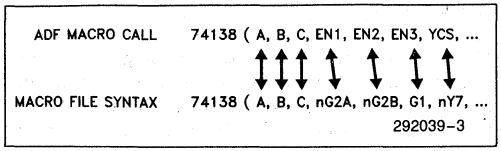
```
74138(A,B,C,nG2A,nG2B,G1,nY7,GND,nY6,nY5,nY4,nY3,nY2,nY1,nY0,VCC)
DEFAULT:(GND,GND,GND,GND,GND,VCC,,GND,,,,,,,,VCC)

NETWORK:

EQUATIONS:

nY0   = !(!A * !B * !C * !nG2A * !nG2B * G1);
nY1   = !(A * !B * !C * !nG2A * !nG2B * G1);
nY2   = !(!A * B * !C * !nG2A * !nG2B * G1);
nY3   = !(A * B * !C * !nG2A * !nG2B * G1);
nY4   = !(!A * !B * C * !nG2A * !nG2B * G1);
nY5   = !(A * !B * C * !nG2A * !nG2B * G1);
nY6   = !(!A * B * C * !nG2A * !nG2B * G1);
nY7   = !(A * B * C * !nG2A * !nG2B * G1);

ENDEF
$
```

292039-2

Figure 2. Sample TTL Macro File (74138.DEV)

```
74138(A,B,C,ENA,ENB,ENC,Y7,GND,Y6,Y5,
Y4,Y3,Y2,Y1,Y0,VCC)
```

In each case, the part name corresponds to the macro part name. The names of the signals differ, but the order of signals match the macro. During processing, the Macro expander assigns node connections between the macro call and the macro file based on the positions of signals, not the names of the signals. For example, note the following macro call to macro file signal assignments:



TTL macro signals originating outside the target EPLD require a prior INPUT macro call in the Network section. All signals used as outputs require a prior OUTPUT macro call in the Network section. Figure 3 shows a sample ADF that uses the 74138 macro. Each input is listed in the INPUTS: declaration and has an INPUT macro call. Outputs are listed in the OUTPUTS: declaration and have OUTPUT macro calls. (EPLD INP and CONF primitive statements may also be used in place of INPUT and OUTPUT macro calls, if desired.)

Gate arrays support a much richer selection of input and output types than EPLDs. Gate array signals originating outside the target gate array device require the appropriate gate array input or output macro calls. When using gate array macros with EPLDs, the I/O macros are implemented in terms of EPLD primitives. Note that when designs targeted for gate arrays are partitioned for multiple EPLDs, many internal gate array signals are transformed into EPLD input and output signals. These signals must be supported by INPUT and OUTPUT macro calls.

Macro Expansion

The Macro Expander identifies and expands each macro call in an ADF with the corresponding macro definition from macro libraries (the TTL library in the case of the 74138). The Macro Expander searches libraries in the following order and in the directories listed:

- MACRO.LIB—first in the current directory, then in other directories specified by the DOS "PATH" variable.
- user libraries (filename.LIB)—names for user libraries are specified in the "IPLS" environment variable. If a pathname and filename are both specified (SET IPLS=C:\MACLIB\USR1.LIB;), the path is treated as an absolute path. If a filename alone is specified (set IPLS=USR1.LIB;), the Macro Expander searches for that library in the directories specified by the "PATH" variable. (IPLS can be set in an AUTOEXEC.BAT file.)
- TTL macro library (TTL.LIB)—first in the current directory, then in other directories specified by the DOS "PATH" variable.

```
YOUR NAME
YOUR COMPANY
DATE
1
A
5C060
One-of-Eight Decoder

OPTIONS: TURBO=OFF
PART: 5C060
INPUTS: A,B,C,G2A,G2B,G1
OUTPUTS: Y7,Y6,Y5,Y4,Y3,Y2,Y1,Y0

NETWORK:

INPUT(A,A)
INPUT(B,B)
INPUT(C,C)
INPUT(G2A,G2A)
INPUT(G2B,G2B)
INPUT(G1,G1)
OUTPUT(Y7,Y7)
OUTPUT(Y6,Y6)
OUTPUT(Y5,Y5)
OUTPUT(Y4,Y4)
OUTPUT(Y3,Y3)
OUTPUT(Y2,Y2)
OUTPUT(Y1,Y1)
OUTPUT(Y0,Y0)
74138(A,B,C,G2A,G2B,G1,Y7,GND,Y6,Y5,Y4,Y3,Y2,Y1,Y0,VCC)

END$
```

292039-4

Figure 3. ADF File Calling the 74138 Macro

- EPLD Custom macro library (EPLDMAC.LIB)—first in the current directory, then in other directories specified by the DOS "PATH" variable.
- reserved library (INTEL.LIB)—first in the current directory, then in other directories specified by the DOS "PATH" variable.

Only the first occurrence of a macro is used. The names TTL.LIB, EPLDMAC.LIB, and INTEL.LIB are reserved by Intel. They may not be used for user libraries and may not be specified in the "IPLS" variable. The "IPLS" variable can contain more than one library name. Each library can have an absolute path or can rely on the "PATH" variable to determine the search path.

The Macro Expander uses the ADF Network and Equation entries from the macro libraries and assigns the appropriate primitives for INPUT and OUTPUT calls. INP primitives are assigned to replace the INPUT macro calls. The OUTPUT calls are assigned primitives with output pins and output enables are supplied where needed.

Combination of primitives is automatically performed when needed. For example, when a feedback primitive such as a NORF feeds an output primitive such as a RONF, the Macro Expander combines the two primitives into a RORF. Combination of primitives conserves resources and results in the shortest possible delay path through the device.

During macro expansion, unused nodes are eliminated. For example, the VCC and GND nodes that correspond to TTL power and ground pins are eliminated. If an input node is not connected to a node in the ADF, the default value for that node is assigned from the

```
NETWORK:

%    INPUT(A,A)    %
%^% A=INP(A)
%    INPUT(B,B)    %
%^% B=INP(B)
%    INPUT(C,C)    %
%^% C=INP(C)
%    INPUT(G2A,G2A)    %
%^% G2A=INP(G2A)
%    INPUT(G2B,G2B)        %
%^% G2B=INP(G2B)
%    INPUT(G1,G1)    %
%^% G1=INP(G1)
%    OUTPUT(Y7,Y7)    %
%^% Y7=CONF(Y7,VCC)
%    OUTPUT(Y6,Y6)    %
%^% Y6=CONF(Y6,VCC)
%    OUTPUT(Y5,Y5)    %
%^% Y5=CONF(Y5,VCC)
%    OUTPUT(Y4,Y4)    %
%^% Y4=CONF(Y4,VCC)
%    OUTPUT(Y3,Y3)    %
%^% Y3=CONF(Y3,VCC)
%    OUTPUT(Y2,Y2)    %
%^% Y2=CONF(Y2,VCC)
%    OUTPUT(Y1,Y1)    %
%^% Y1=CONF(Y1,VCC)
%    OUTPUT(Y0,Y0)    %
%^% Y0=CONF(Y0,VCC)
%    74138(A,B,C,G2A,G2B,G1,Y7,GND,Y6,Y5,Y4,Y3,Y2,Y1,Y0,VCC)    %

EQUATIONS:

%^% Y0=!(!A*!B*!C*!G2A*!G2B*G1);
%^% Y1=!(A*!B*!C*!G2A*!G2B*G1);
%^% Y2=!(!A*B*!C*!G2A*!G2B*G1);
%^% Y3=!(A*B*!C*!G2A*!G2B*G1);
%^% Y4=!(!A*!B*C*!G2A*!G2B*G1);
%^% Y5=!(A*!B*C*!G2A*!G2B*G1);
%^% Y6=!(!A*B*C*!G2A*!G2B*G1);
%^% Y7=!(A*B*C*!G2A*!G2B*G1);
```

292039-5

Figure 4. Network and Equations for 74138.SDF

DEFAULT: section of the macro file. Note, however, that the default value for each input in the macro file may be the value that disables the input or, for data inputs, is usually a logic 0. To be certain of the level used, specify a "VCC" or "GND" in the macro call for unused inputs.

The INPUT and OUTPUT calls and the original macro call are "commented out" by surrounding them with percent (%) signs. The %^% string is placed at the start of lines where primitives are created by the Macro Expander. The fully expanded file is written to the disk using the original filename and a .SDF extension. Figure 4 shows the Network and Equation sections for the 74138 SDF.

One final note with regard to compiling ADFs that use macros. Warning messages are typically encountered while compiling files that use macros. The most common message is "***WARN-XLT-Node Missing Destination". This message is displayed as unused nodes from a macro are deleted. For example, if a macro using a NOCF primitive is combined with a CONF and the original feedback is not needed, the warning is displayed as the feedback is deleted.

Multiple Macro Calls

The Macro Expander allows use of more than one macro in ADFs. Each macro must have its own call, even when the same macro is used more than once.

For example, to implement two 74138s, each case or "instance" must have its own call:

```
74138(A,B,C,G2A,G2B,G1,Y7,GND,Y6,Y5,
Y4,Y3,Y2,Y1,Y0,VCC)

74138(A,B,C,G3A,G3B,G1,YF,GND,YE,YD,
YC,YB,YA,Y9,Y8,VCC)
```

In this example, many of the inputs are routed to both devices. The Macro Expander automatically generates internal nodes for each instance of the macro. Each node is assigned a unique number based on the position of the macro in the Network section (i.e., . . 0140, . . 0141, etc. for nodes connecting to the 14th primitive in the Network section).

For traceability, you can define your own instance names for nodes of different macros by including the instance name in a comment immediately following the macro call. For example, to call two 74161 macros, one as Shift Register A and the other as Shift Register B, enter the calls as follows:

```
74161(CLR,CK,A,B,C,D,ENP,,LD,ENT,QD,
QC,QB,QA,RD1,) % SFA %

74161(CLR,CK,E,F,G,H,ENP,,LD,ENT,QH,
QG,QF,QE,RC2,) % SFB %
```

The Macro Expander uses the first three ASCII characters after the first percent sign (%), except for white space, to create instance numbers. For example, internal nodes for the first three signals of each macro call will be:

```
..SFAN1, ..SFAN2, ..SFAN3,

..SFBN1, ..SFBN2, ..SFBN3
```

where SFA/SFB are the user-defined instance names and N1, N2, N3 are the node numbers associated with each instance. For cases where no internal nodes numbers are generated, the Macro Expander simply ignores the instance name.

Outputs from one macro call can be used as inputs for other calls, as follows:

```
74138(A,B,C,G2A,G2B,G1,Y7,GND,Y6,Y5,
Y4,Y3,Y2,Y1,Y0,VCC)

74138(A,B,C,Y7,G3B,G1,YF,GND,YE,YD,YC,
YB,YA,Y9,Y8,VCC)
```

Here the Y7 output from the first decoder feeds an enable input of the second decoder.

Different macros are connected in the same manner. For example, the following macro calls connect the outputs from a 74138 decoder to the inputs of 74175 latches:

```
74138(A,B,C,G2A,G2B,G1,Y7,GND,Y6,Y5,
Y4,Y3,Y2,Y1,Y0,VCC)

74175(CLR,0Q,n0Q,Y0,Y1,n1Q,1Q,GND,CLK,
2Q, n2Q,Y2,Y3,n3Q,3Q,VCC)

74175(CLR,4Q,n4Q,Y4,Y5,n5Q,5Q,GND,CLK,
6Q, n6Q,Y6,Y7,n7Q,7Q,VCC)
```

Each decoder output is routed to a 74175 input. The 74175 macro produces both true and complement latched outputs.

Guidelines/Pitfalls

The following paragraphs discuss some general guidelines for using macros:

- Because the Macro Expander supports only one level of hierarchy, there is a tendency for p-terms to multiply quickly when several macros are connected together. In many cases, the total number of p-terms exceeds the capacity of the target EPLD. One method of avoiding problems with excessive p-terms is to route the outputs from a macro function through EPLD macrocells and use the feedbacks from the macrocells as inputs to the subsequent macro functions. This partitioning of functions trades off device resources for a lower p-term count.

- Implementation of some TTL macros requires primitives that are not supported on all devices. The .DOC file for a device notes any device dependency. In many cases, a modification to the basic TTL functions results in device independence. For example, a NOCF, which is not supported on all EPLDs, can be changed to a COIF, which is supported on all devices.
- Some macros use primitives that specify an output pin (COIF, CONF, RORF, etc.). These primitives must be supported with a signal name in the OUTPUTS: declaration and by an OUTPUT call in the Network Section of the ADF. Failure to provide this support causes the following error message during compilation:

```
***ERR-XLT-undeclared output name
```

If you encounter this error, check the macro file for output primitives that require ADF support.

Macro Usage Summary

ADF macro calls must observe the following guidelines:

- Macros are called from the Network Section of an ADF.
- The name in the call must match the name in the macro file (e.g., 74138 = 74138).
- All input and output pins on the target device must have both: (1) a corresponding signal name in the INPUTS: or OUTPUTS: declaration, and (2) a corresponding INPUT or OUTPUT macro call in the Network section. It is recommended that the same node name be used on both sides of each INPUT and OUTPUT macro call. This is required when macros containing CONFs are used. (EPLD INP and CONF primitives may also be used).
- All INPUT and OUTPUT calls in the Network section must precede any other macro call.
- Node connections within an ADF are made based on the names of the nodes.
- Connections between the macro call and macro files are based on the position of signal names in the call. Therefore, the sequence of inputs and outputs in a macro call must match the sequence of inputs and outputs in the corresponding macro file.

EXAMPLE 1: TTL MACROS

This section provides an example design using TTL macros.

Circuit

The design is a two-stage decoder using a 74138 macro and two 74139 macros. Figure 5 shows the schematic for the circuit. Each 74139 macro represents one half of a TTL 74139 device. Note that two of the outputs from the 74138 are routed back to enable the two 74139 decoders.

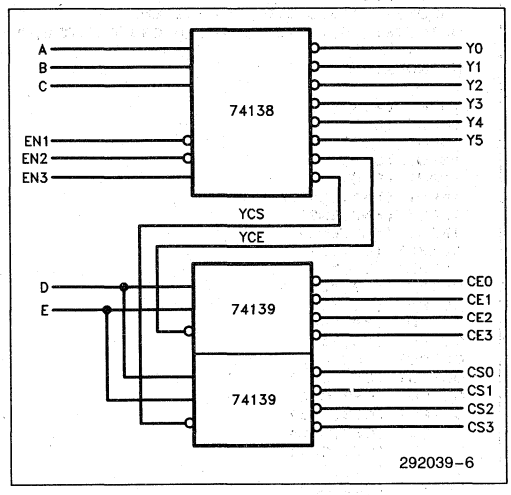


Figure 5. Schematic Diagram for Two-Stage Decoder

Figure 6 shows the ADF file containing the macro calls that implement the circuit. The two internal feedback signals (YCS and YCE) do not show up in the INPUTS: or OUTPUTS: declarations and are not represented by INPUT or OUTPUT calls in the Network section. The sequence of signals in the INPUTS: and OUTPUTS: declarations of the ADF is not important.

In the NETWORK: section, however, order is important. INPUT and OUTPUT calls must be listed before any other macro calls. This is a requirement of the Macro Expander. The sequence of signals within the ADF macro call is **critical**, as the Macro Expander automatically assigns macro call signals to macro file signals based on position.

Internal connections between macros are established by assigning the same name to the respective signals. For example, YCS in the 74138 macro call in Figure 7 represents the nY6 output from the 74138, while YCS in the 74139 macro call represents the 1G input to one 74139 decoder. Use of the same name establishes the connection. In the same manner, use of the signal name YCE connects the nY7 output from the 74138 to the 1G input of the second 74139.

```
DANIEL E. SMITH
INTEL CORPORATION
2/27/87
1
A
5C090
TWO-STAGE DECODER

OPTIONS: TURBO=OFF
PART: 5C090
INPUTS:  A,B,C,D,E,EN1,EN2,EN3
OUTPUTS: YO,Y1,Y2,Y3,Y4,Y5,CSO,CS1,CS2,CS3,CEO,CE1,CE2,CE3

NETWORK:

INPUT (A,A)
INPUT (B,B)
INPUT (C,C)
INPUT (D,D)
INPUT (E,E)
INPUT (EN1,EN1)
INPUT (EN2,EN2)
INPUT (EN3,EN3)
OUTPUT (YO,YO)
OUTPUT (Y1,Y1)
OUTPUT (Y2,Y2)
OUTPUT (Y3,Y3)
OUTPUT (Y4,Y4)
OUTPUT (Y5,Y5)
OUTPUT (CSO,CSO)
OUTPUT (CS1,CS1)
OUTPUT (CS2,CS2)
OUTPUT (CS3,CS3)
OUTPUT (CEO,CEO)
OUTPUT (CE1,CE1)
OUTPUT (CE2,CE2)
OUTPUT (CE3,CE3)
74138(A,B,C,EN1,EN2,EN3,YCS,GND,YCE,Y5,Y4,Y3,Y2,Y1,YO,VCC)
74139(YCS,D,E,CSO,CS1,CS2,CS3,GND,VCC)
74139(YCE,D,E,CEO,CE1,CE2,CE3,GND,VCC)

END$
```

292039-7

Figure 6. ADF File for Two-Stage Decoder Using TTL Macros

Sample Session

This session assumes familiarity with the iPLS II Logic Optimizing Compiler (LOC). For detailed information on the LOC, refer to Chapter 4 of the *iPLS II User's Guide,* order number: 450196. Proceed as follows to implement the TTL macro design shown here:

1. Use a standard ASCII text editor to create the ADF shown in Figure 7 under the name DECODE.ADF.
2. Invoke the iPLS II Menu by entering:

```
IPLS <Enter>
```

3. Invoke the LOC from the Main Menu by pressing <F4>.
4. Answer the LOC promts as follows:

```
Input Format?          <Enter>
File Name?             DECODE <Enter>
Minimization?          Y
Inversion Control?     N
LEF Analysis?          Y
Error Message File     <Enter>
```

The LOC then asks:

```
Do you wish to run under the above
conditions [Y/N]?
Enter: Y
```

The LOC expands the macros and compiles the expanded file to produce a JEDEC programming file (DECODE.JED), a utilization report file (DECODE. RPT), a minimized equation file (DECODE. LEF), and an error message file (DECODE.ERR). For tracability, a file called DECODE.SDF is created to show the expanded form of the ADF output by the Macro Expander.

5. The LOC terminates execution with the following message:

```
LOC cycle successfully completed
```

You can examine the LEF file to see the minimized form of the design. The LEF shows the EPLD primitives used to implement the design. Macro calls are not shown. If you wish, you can also use LPS (Logic Programmer Software) to program a part.

EXAMPLE 2: MIXING MACROS AND EPLD PRIMITIVES

This final example uses TTL macros together with standard EPLD primitives.

Circuit

The example circuit here is the 74138 macro used in example 1 with two of the outputs routed through additional combinatorial logic and RONF (Registered Output — No Feedback) primitives. Figure 7 shows the circuit. CS2 and CS3 are qualified by two additional inputs (RD* and WR*) to set or clear two latches. This is a configuration commonly used in microcomputer systems, where control signals are set and reset based on the address and command signals but not on a data value. A read to the port decoded by CS2 sets output LCS2 (Latched CS2) high. A write to that same port clears LCS2 low.

Figure 8 shows the ADF that implements the example circuit. This is the same ADF used in Figure 6, with the addition of several primitives and equations. The data inputs to both latches are tied to VCC. When RD* and the chip enable are both low, the respective clock signal goes low. As RD* or chip enable go high, the rising edge of the clock signal triggers the register, driving the output high.

Note that many Intel EPLDs do not support multiple product terms for register clocks. Therefore, the clock buffer primitive is driven by a macrocell configured as a COIF (Combinatorial Output—Input Feedback). Control signals (Clear and Preset) for many EPLDs also support only one product term. In this case, however, the NOR gate driving the clear input to the RONFs can be minimized to a single p-term. Thus a low on WR* and chip enable clears the respective latch to logic 0. (The intermediate macrocell for the Read function can be omitted for EPLDs that support two p-terms on register clocks.)

The connections between the TTL macros and the EPLD primitive are made by assigning the appropriate names to the input and output nodes. The CS2 and CS3 signals from the first example are no longer outputs, but are simply inputs to equations that feed the LCS2 and LCS3 RONF primitives.

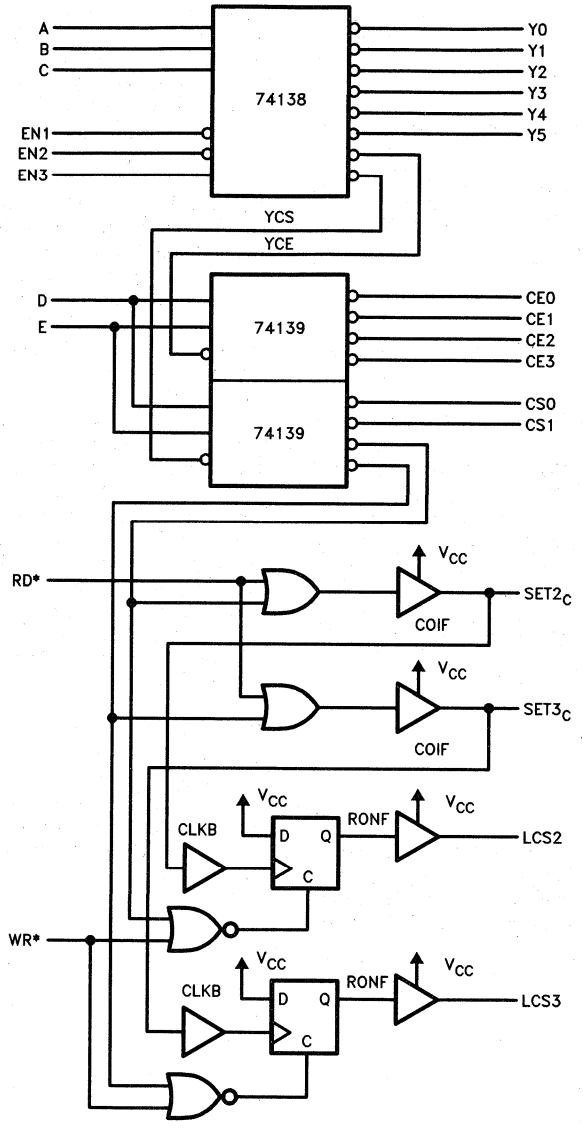


Figure 7. Schematic of Decoder Circuit with Latched Outputs

```
DANIEL E. SMITH
INTEL CORPORATION
2/27/87
1
A
5C090
DECODER WITH TWO LATCHED OUTPUTS

OPTIONS:  TURBO=OFF
PART:  5C090
INPUTS:  A,B,C,D,E,EN1,EN2,EN3,RD*,WR*
OUTPUTS:  SET2c,SET3c,Y0,Y1,Y2,Y3,Y4,Y5,CS0,CS1,LCS2,LCS3,CE0,CE1,CE2,CE3

NETWORK:

INPUT (A,A)
INPUT (B,B)
INPUT (C,C)
INPUT (D,D)
INPUT (E,E)
INPUT (EN1,EN1)
INPUT (EN2,EN2)
INPUT (EN3,EN3)
OUTPUT (Y0,Y0)
OUTPUT (Y1,Y1)
OUTPUT (Y2,Y2)
OUTPUT (Y3,Y3)
OUTPUT (Y4,Y4)
OUTPUT (Y5,Y5)
OUTPUT (CS0,CS0)
OUTPUT (CS1,CS1)
OUTPUT (CE0,CE0)
OUTPUT (CE1,CE1)
OUTPUT (CE2,CE2)
OUTPUT (CE3,CE3)
74138(A,B,C,EN1,EN2,EN3,YCS,GND,YCE,Y5,Y4,Y3,Y2,Y1,Y0,VCC)
74139(YCS,D,E,CS0,CS1,CS2,CS3,GND,VCC)
74139(YCE,D,E,CE0,CE1,CE2,CE3,GND,VCC)
RD = INP(RD*)
WR = INP(WR*)
LCS2 = RONF(VCC,SET2,CLR2,GND,VCC)
LCS3 = RONF(VCC,SET3,CLR3,GND,VCC)
SET2 = CLKB(SET2c)
SET3 = CLKB(SET3c)
SET2c,SET2c = COIF(ST2,VCC)
SET3c,SET3c = COIF(ST3,VCC)

EQUATIONS:

ST2 = RD + CS2;
CLR2 = /(WR + CS2);
ST3 = RD + CS3;
CLR3 = /(WR + CS3);

END$
```

292039-12

Figure 8. ADF File for Decoder with Latched Outputs

Sample Session

To implement this ADF in an actual session, follow the steps described for Example 1, substituting the name LDECODE for DECODE. iPLS II produces a JEDEC programming file (LDECODE.JED), a utilization report file (LDECODE.RPT), a minimized equation file (LDECODE.LEF), and an error message file (LDECODE.ERR). For traceability, a file called LDECODE.SDF is created to show the expanded form of the ADF output by the Macro Expander.

APPLICATION NOTE

AP-312

October 1988

Creating Macros for EPLD Designs

DANIEL E. SMITH
PROGRAMMABLE LOGIC APPLICATIONS
INTEL CORPORATION

Order Number: 292040-002

CREATING MACROS FOR EPLD DESIGNS

CONTENTS

4

INTRODUCTION

The iPLS II (Intel Programmable Logic Software II) Logic Optimizing Compiler includes a Macro Expander that supports the use of macros in EPLD designs. These macros can include TTL and EPLD custom macros available from Intel, or proprietary macros developed by a user. This application note shows how to create user-defined macros and how to build macro libraries with Intel's Macro Librarian, an optional software package for use with iPLS II. A design example also shows creation of a user-defined macro and its use in an ADF (Advanced Design File). Detailed information on using the TTL Macros in iPLS II ADFs are described in a companion application note, AP-311 "Using Macros in EPLD Designs", Order Number: 292039. This application note concentrates on creating macros; it assumes that you have read and understood the discussion on using macros in AP-311.

OVERVIEW

iPLS II allows designers to include macro calls in design files to implement common circuit functions. Macros calls are subsequently expanded by the LOC (Logic Optimizing Compiler) into the ADF network and/or equation entries required to perform the desired functions. Macros can be connected together or used in conjunction with standard iPLS II EPLD primitives.

By following the macro file format described in this note, users can also create their own proprietary macros with an ASCII text editor. These macro files can then be stored in user-defined libraries by using Intel's Macro Librarian software. User-defined macros can be called from ADFs created by a text editor or by schematic capture software that supports user-defined symbols and that outputs in ADF format. User-defined macros can optimize development of EPLD designs by modularizing the design process and by allowing the design process to proceed at a higher level than with EPLD primitives alone. iPLS II support for user-defined macros (see in Figure 1) includes the following:

- MLIB, the optional iPLS II Macro Librarian for creating macro libraries from individual user-defined macro files.
- a Macro Expander in the LOC that expands macro calls in ADFs with the contents of the corresponding macros from libraries.

This application note describes how to create macro files, store them in libraries with MLIB, and shows how to call them from ADFs created by a text editor. *For information on creating user-defined macro symbols with schematic capture packages, refer to the appropriate manual for the schematic capture package you are using.* SCHEMA II-PLD available from Intel supports user-defined symbols and outputs in ADF format.

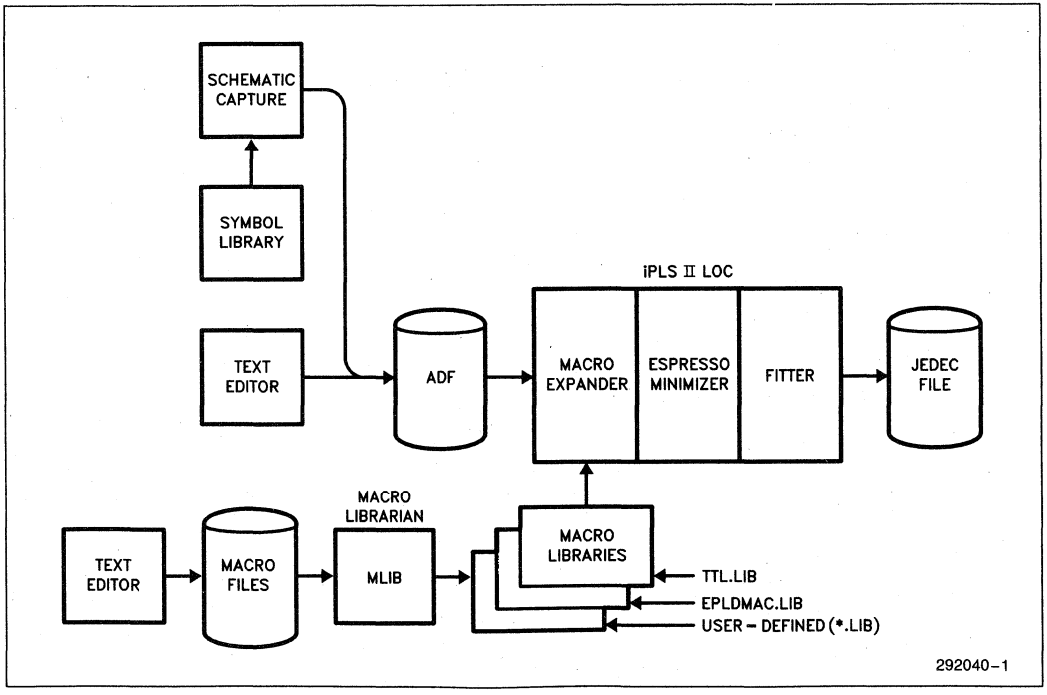


Figure 1. Macro Support for iPLS II

(SCHEMA II-PLD is based on SCHEMA II from Omation, Inc. The Intel EPLD Design Manager, also available from Intel, allows existing SCHEMA II users to design with EPLDs and macros.)

MACRO FILES

This section describes iPLS II macro files. **User-defined macro files must follow the guidelines presented here** to be successfully processed by the Macro Librarian (MLIB) and expanded by the iPLS II LOC Macro Expander.

Macro filenames follow DOS conventions. It is recommended that macro filenames end with the extension .DEV, which is the default for MLIB. Only one macro can be contained in a macro file. Macro files are comprised of three sections:

- Header
- Network Section
- Equation Section

All macro files must end with the literal "ENDEF". Figure 2 shows a sample macro file for a proprietary part (16207), a "black box" containing random logic.

```
16207(A,B,C,D,E,F,U,V,W,X,Y,Z)
DEFAULT:(GND,GND,GND,VCC,VCC,VCC,,,,,,)

EQUATIONS:
        U = /(A * B);
        V = /(/E * A * B);
        W = /(D * C * A * /E);
        X = /(/D * E);
        Y = /(F * D * A);
        Z = F * /E;

ENDEF
                                        292040-2
```

Figure 2. Sample Macro File for "Black Box" (16207.DEV)

Header

Headers for macro files contain two lines. The first line includes the name of the macro function and a list of inputs and outputs for the macro. The second line contains defaults for the device.

The name of the macro can be a device number (16207, 83546, etc.), function name (ADDRCNT, CMDLO, etc.), or any name up to eight characters long. No spaces or comments precede the name.

Inputs and Outputs follow immediately after the macro name and are enclosed in parentheses. I/O signal names may be up to eight characters long, but may not contain pin numbers. For user-defined macros, signals may be listed in any order desired. For example, any of the following entries are legal:

16207 (A,B,C,D,E,F,U,V,W,X,Y,Z)

16207 (B,D,A,R,Z,U,W,C,F,X,E,Y)

16207 (Z,Y,X,W,V,U,F,E,D,C,B,A)

Note that this **first line of the header forms the template used to call the Macro from the ADF.** The Macro Expander connects ADF nodes in the macro call to I/O signals in the macro file on the basis of **position**, not on the basis of node name.

The second line in the header specifies defaults for inputs (VCC or GND) in cases where those signals are left unconnected. The DEFAULT: line must be included in the macro definition file, even when no defaults are used in the ADF. The keyword DEFAULT: is the first entry in this line. The default values for all signals follow immediately and are enclosed in parentheses. Input defaults may be VCC or GND. The position of the default value corresponds to the signal listed in the previous line.

Defaults for outputs are blank, but a comma (,) must be present (place holder) for each output signal except the last. For example, the 16207 black box contains six inputs (A through F) and six outputs (U through Z). The first two lines for this macro might be:

```
16207 (A,B,C,D,E,F,U,V,W,X,Y,Z)
DEFAULT: (GND,GND,GND,VCC,VCC,VCC,,,,,,)
```

Defaults for inputs A through C are GND; defaults for inputs D through F are VCC. Defaults for the outputs are not specified, but the comma denotes the positions for those signals.

Defaults should be chosen with care. Clears, Presets, Loads, etc. should be disabled in most cases. Enables should be enabled. Input defaults can also be left blank as long as those inputs are connected to nodes in the ADF that calls the macro, but it is recommended that they be specified in the macro file.

Network Section

The NETWORK: section lists the EPLD primitives used to implement the desired functions. The Network Section follows ADF syntax rules. As far as possible, the macros should be implemented in equations to eliminate concern about feedbacks and output enables. In the case of a circuit that requires macrocell registers, the feeback-only form of the primitive should be used so that the Macro Expander can make the correct pin connections. The following example shows this:

```
OUT1 = NORF (INd,CLK,GND,GND)
```

During processing, the Macro Expander connects the feedback to an output (if necessary) and supplies the required output enable node name. The Macro Expander also eliminates unneeded Network and Equations entries if they are not used by an ADF.

If no network entries are required (i.e., a macro implemented entirely in equations), the entire Network section may be omitted, including the keyword NETWORK:. In many cases, equations alone can implement the desired functions.

Equations Section

The EQUATIONS: section lists the Boolean equations for the desired functions and follows ADF syntax rules, with one exception; intermediate equations are not permitted in macro files. If no equation entries are required (i.e., a macro implemented entirely in the Network Section), the entire Equation section may be omitted, including the keyword EQUATIONS:.

Comments and White Space

Comments can be placed anywhere in a macro file except before the name and signals on the first line. Comments must be enclosed in percent signs, as follows:

```
% THIS IS A SAMPLE COMMENT %
```

White space can appear on any line except the first two lines.

MACRO LIBRARIAN

The Macro Librarian (MLIB) is an optional software package that combines individual macro files into macro libraries. These libraries are in turn used by the LOC Macro Expander. MLIB can be invoked from the command line, from command files, or from a combination of both. Figure 3 shows a block diagram of the Macro Librarian.

Syntax for MLIB command lines is as follows:

MLIB [-options] [@cmdfile] [file1 file2 . . .]
<Enter>

-d directory. Displays directory information for the library being created.

-v verbose. Print status during processing. When not specified, status messages are suppressed.

-l lib list. Lists the contents of existing macro library to console. This option may not be used while building a library.

-o lib name of the target macro library. MACRO.LIB is the default when no name is specified. TTL.LIB, EPLDMAC.LIB, and INTEL.LIB are reserved for Intel libraries and may not be used.

-s string include version stamp in macro library. The version string can be up to 7 characters long. "V1.00" is the default stamp.

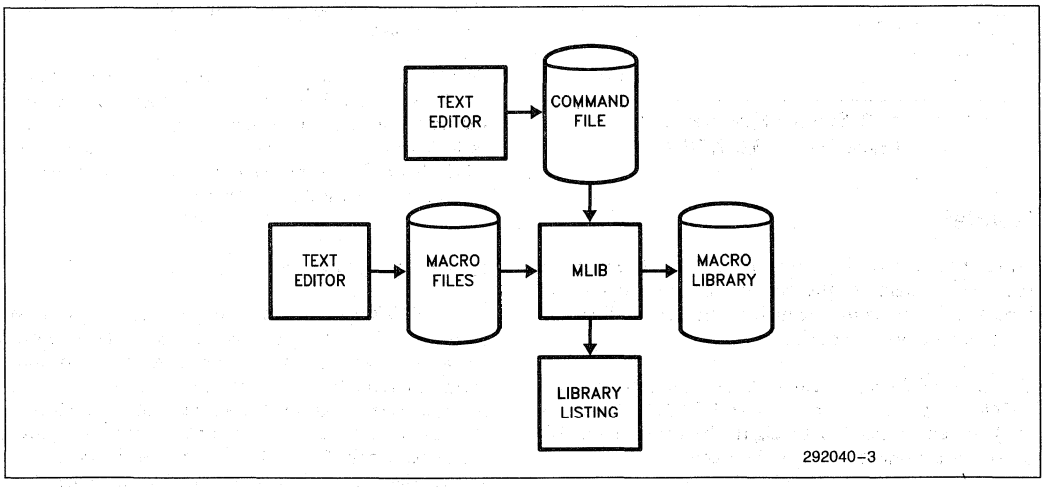


Figure 3. Macro Librarian Block Diagram

-c string include copyright string in macro library. The copyright string can be up to 61 characters long and, if blanks are used, must be contained in quotation marks, for example, "texta textb".

@cmdfile name of command file. The command file can include options and macro filenames. The @ symbol must precede the filename.

file1 . . . name of device files to be included in the macro library. Separate files by spaces.

For example, the following command line:

MLIB -v -s 2.00 -o USER.LIB @USERLIST <Enter>

creates a library called USER.LIB that includes all the individual macro files contained in the command file USERLIST. MLIB displays status messages as it processes the macro files in USERLIST (-v). The library is created as version 2.00 (-s).

Macro library filenames follows DOS conventions and should end with the extension .LIB to be recognized by the Macro Expander. TTL.LIB, EPLDMAC.LIB, and INTEL.LIB are reserved and may not be used.

USERLIST is the name of the command file and must be preceded by the @ symbol. The command file is simply an ASCII text file that can be modified to contain any number of macros desired. MLIB processes the entire list of macros on each invocation. To add a new macro to an existing library, add the name of the macro to USERLIST, and create the new library by entering the command line shown above. Command file names follow DOS conventions. MLIB supplies a .DEV extension if no extension is specified. MLIB searches first in the current directory, then along the DEV environment variable, and finally along the PATH environment variable for the files.

In order to connect inut and output primitives, the files INPUT.DEV and OUTPUT.DEV must be included in at least one of the libraries. These files are contained in the TTL macro library.

Figure 4 shows a sample MLIB command file that includes options, the library name, and the names of seven macro files to be included in the library in addition to the INPUT and OUTPUT macros. The format of the command file is free form. Note that comments can be included in the command file and must be contained within percent (%) signs.

Note that the -1 option cannot be included in an MLIB command file; it can only appear on the command line. The -1 option lists the contents of existing libraries; it does not list library contents while building a library.

```
-o PROJA.LIB % macro library name %
-v
-s V1.50 % version number %
-c "Copyright  (C)  Date, Your Company, Your Name"
                % copyright information %
-d % display directory %

% include the following macros %

INPUT.DEV    OUTPUT.DEV    7408.DEV
7487.DEV     74138.DEV     74139.DEV
74151.DEV    74157.DEV     74251.DEV
```

292040-4

Figure 4. Sample Command File for MLIB

The command line to process the file shown in Figure 4 is as follows:

MLIB @SAMPLE <Enter>

where SAMPLE is the name of the command file.

To list the contents of PROJA.LIB after creation, invoke MLIB as follows:

MLIB -l PROJA.LIB

This command line lists the macros in PROJA.LIB to the screen. The DOS file redirection capability can also be used to create a disk file listing the contents of macro libraries. For example:

MLIB -l PROJA.LIB > PROJA.DOC

4

SAMPLE SESSION: COMMAND DECODER USING MACROS

Decoding logic is one common function implemented by programmable logic devices. The target circuit for this example is a device that decodes microprocessor command signals in selected address ranges. The target application and decoder requirements are as follows:

- The target application is a 16-bit microcomputer system with 1-Megabyte of memory and about two dozen I/O ports.
- The memory is divided into shared memory (lower 512K bytes) and local memory (upper 512K bytes). Shared memory resides off the processor board and requires active low memory command signals. Local memory resides on-board and requires active high memory command signals.
- I/O ports are also split between on-board devices requiring active high signals and off-board devices requiring active low signals. I/O devices between the address range F000-FFFFH are on-board; devices below that range (0000-EFFFH) are off-board.

- All interrupt requests are resolved by an on-board interrupt controller. Therefore, only an active high on-board interrupt acknowledge signal is needed.
- On-board control signals are always high or low, never three-stated. Off-board control signals are three-stated when not being used to execute a bus cycle. An external bus arbiter accepts a request signal from the command decoder and, after gaining control of the bus, sends address enable and command enable signals back to the command decoder.

Figure 5 shows a block diagram of the application, including the target EPLD design. The three functional blocks to be included in the EPLD are highlighted (not shaded).

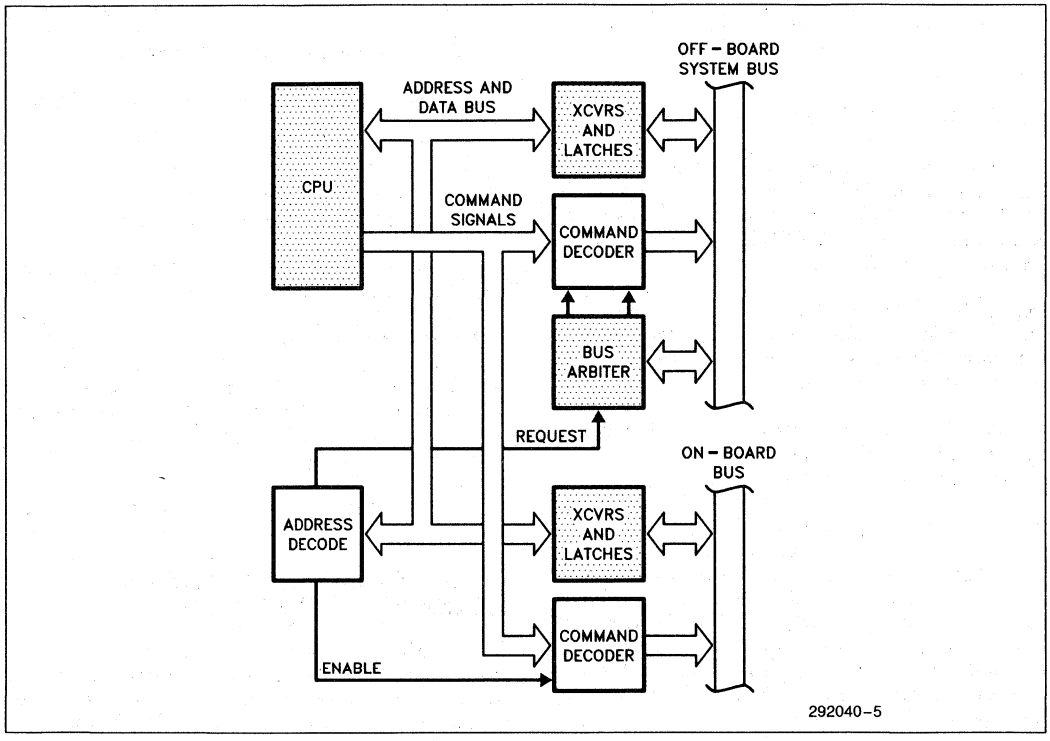


Figure 5. Block Diagram of Target Circuit and Application

Creating the Macro

Figure 6 shows a schematic diagram for the active low command decoder implemented with OR gates (low inputs enable the outputs; high inputs disable the outputs). Figure 7 shows the macro file that implements the circuit (CMDLO.DEV). This file was created with an ASCII text editor. Used as is, it provides the active low outputs for the design. With inputs RD, WR, and INTAIN inverted, it also provides the active high outputs for the design. This design uses CONF primitives to implement the three-state outputs in the macro. As an alternative, equations alone could have been used with the CONFs included in the ADF.

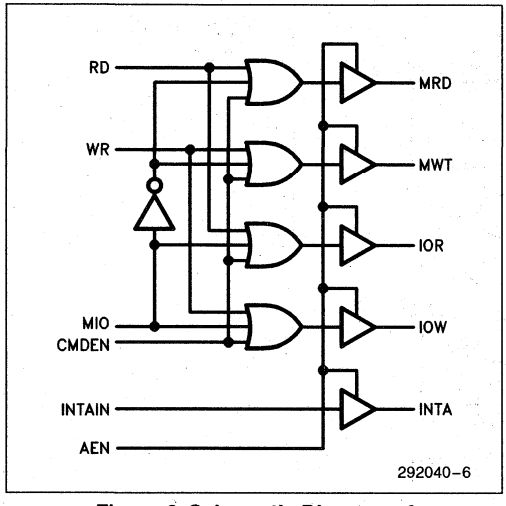


Figure 6. Schematic Diagram of Command Decoder

Building the Library

Use your text editor to create an MLIB command file that includes CMDLO.DEV, INPUT.DEV, and OUTPUT.DEV. The following example shows a sample command file named MACLIST.

```
-v  % show status %
-c  "1987, AP-312 Sample Macro Library"
-o  AP312.LIB
-d  % show the list %

% include the following macros %

CMDLO.DEV  INPUT.DEV  OUTPUT.DEV
```

Invoke the Macro Librarian with the following command line:

```
MLIB  @MACLIST
```

The Macro Librarian processes the three macro files and stores them in a user library named AP312.LIB. The library contains the copyright statement "1987, AP-312 Sample Macro Library". When processing is complete, MLIB returns control to DOS.

Creating the ADF

Figure 8 shows a schematic diagram for the target circuit. Figure 9 shows the ADF for the circuit (COMCODE.ADF), which invokes both instances of the CMDLO macro and contains equations used to enable the decoders under the proper conditions. The ADF signal named ONBEN (On-Board Enable) enables the active high decoder). The AEN (Address Enable) input to the on-board decoder is left unconnected. The default (always enabled) will be used.

```
CMDLO(MIO,RD,WR,INTAIN,CMDEN,AEN,MRD,MWT,IOR,IOW,INTA)
DEFAULT:(GND,VCC,VCC,VCC,GND,GND,,,,,)

NETWORK:

MRD  = CONF(MRDc,AEN)
MWT  = CONF(MWTc,AEN)
IOR  = CONF(IORc,AEN)
IOW  = CONF(IOWc,AEN)
INTA = CONF(INTAIN,AEN)

EQUATIONS:

MRDc = /MIO + RD  + CMDEN;
MWTc = /MIO + WR  + CMDEN;
IORc = MIO  + RD  + CMDEN;
IOWc = MIO  + WR  + CMDEN;

ENDEF
```

292040-7

Figure 7. Macro File for Command Decoder (CMDLO.DEV)

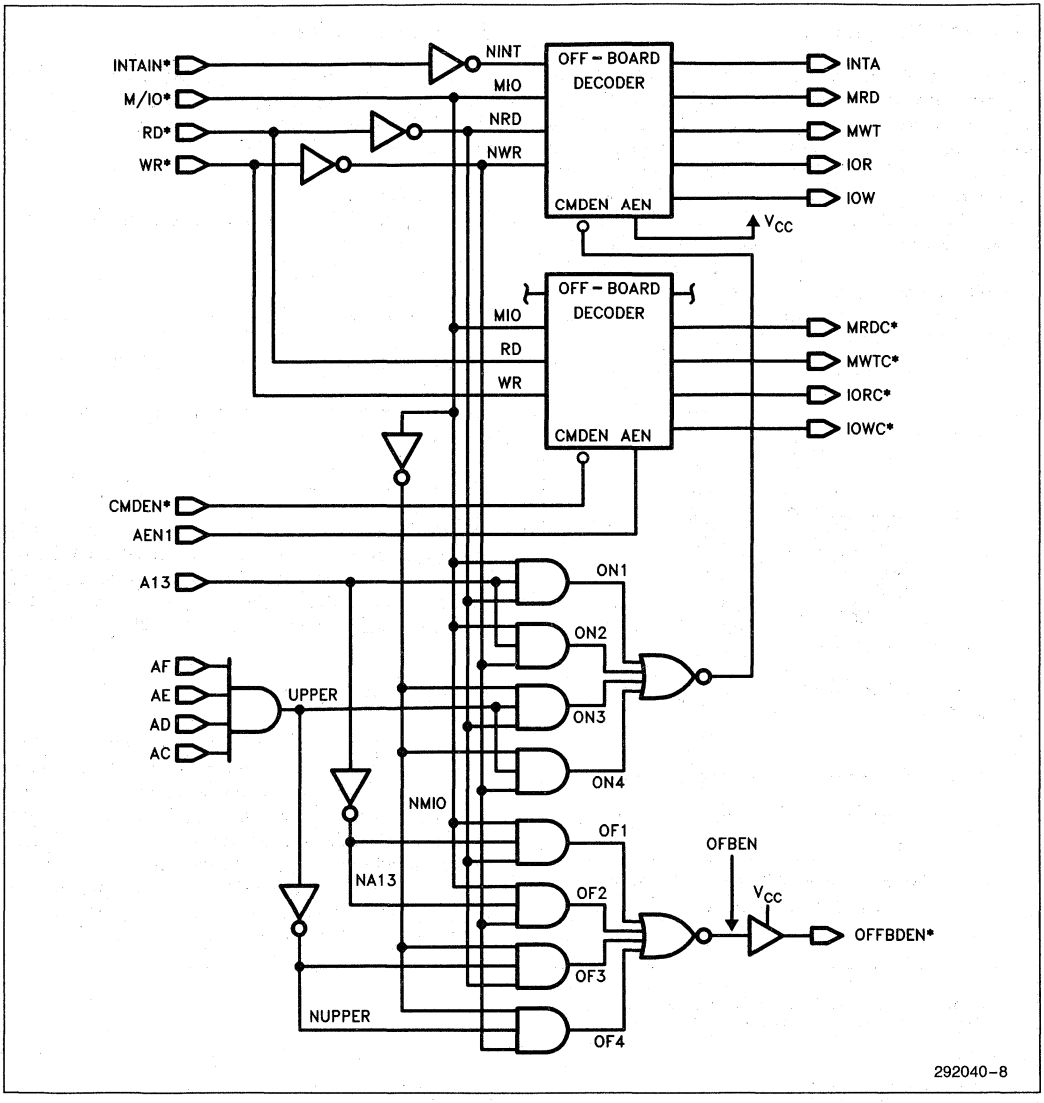


Figure 8. Schematic Diagram for COMCODE.ADF

```
DANIEL E. SMITH
INTEL CORPORATION
4/7/87
1
A
16209-001
COMMAND DECODER

OPTIONS:  TURBO=ON
PART:  5C090
INPUTS:  MIO, RD, WR, INTAIN, CMDEN, AEN1, A13, AF, AE, AD, AC
OUTPUTS:  MRD, MWT, IOR, IOW, INTA, MRDC, MWTC, IORC, IOWC, OFFBDEN

NETWORK:

INPUT(MIO,MIO)
INPUT(RD,RD)
INPUT(WR,WR)
INPUT(INTAIN,INTAIN)
INPUT(CMDEN,CMDEN)
INPUT(AEN1,AEN1)
INPUT(A13,A13)
INPUT(AF,AF)
INPUT(AE,AE)
INPUT(AD,AD)
INPUT(AC,AC)

OUTPUT(MRD,MRD)
OUTPUT(MWT,MWT)
OUTPUT(IOR,IOR)
OUTPUT(IOW,IOW)
OUTPUT(INTA,INTA)
OUTPUT(MRDC,MRDC)
OUTPUT(MWTC,MWTC)
OUTPUT(IORC,IORC)
OUTPUT(IOWC,IOWC)

CMDLO(MIO,RD,WR,,CMDEN,AEN1,MRDC,MWTC,IORC,IOWC,) % OFB %

CMDLO(MIO,NRD,NWR,NINT,ONBEN,VCC,MRD,MWT,IOR,IOW,INTA) % ONB %

OFFBDEN = CONF(OFBEN,VCC)
OFBEN   = NOR(OF1,OF2,OF3,OF4)
ONBEN   = NOR(ON1,ON2,ON3,ON4)
NRD     = NOT(RD)
NWR     = NOT(WR)
NINT    = NOT(INTAIN)
NMIO    = NOT(MIO)
NUPPER  = NOT(UPPER)
NA13    = NOT(A13)

EQUATIONS:

UPPER = (AF * AE * AD * AC);
ON1   = (MIO * A13 * NRD);
ON2   = (MIO * A13 * NWR);
ON3   = (NMIO * UPPER * NRD);
ON4   = (NMIO * UPPER * NWR);
OF1   = (MIO * NA13 * NRD);
OF2   = (MIO * NA13 * NWR);
OF3   = (NMIO * NUPPER * NRD);
OF4   = (NMIO * NUPPER * NWR);

END$
```

292040-9

Figure 9. ADF for COMCODE.ADF

OFFBEN (Off-Board Enable) requests permission to access the off-board bus from the external bus arbiter. The bus arbiter enables the off-board decoder via AEN1 (Address Enable 1) and CMDEN (Command Enable). CMDEN allows the appropriate signal to go high or low, and AEN1 causes the outputs to independently enter or exit a high impedance state (three-state).

Note the same name is used for both nodes of each INPUT and OUTPUT macro call. Use of the same name ensures proper connection when the Macro Expander eliminates redundant primitives (for example, a CONF feeding another CONF).

Compiling the Design

Proceed as follows to compile the ADF.

1. Include AP312.LIB in the IPLS environment variable. From the DOS command prompt, type:

   SET IPLS=C:\IPLSII\AP312.LIB; . . . <Enter>

   For user-defined macro libraries that are regularly accessed, the IPLS variable can be set in an AUTOEXEC.BAT file.

2. Invoke the iPLS II Menu by entering:

   IPLS <Enter>

3. Invoke the LOC from the Main Menu by pressing <F4>.

4. Answer the LOC prompts as follows:

Input Format?	<Enter>
File Name?	COMCODE <Enter>
Minimization?	Y
Inversion Control?	N
LEF Analysis?	Y
Error Message File	COMCODE.ERR <Enter>

   The LOC then asks:

   Do you wish to run under the above conditions [Y/N]?

   Enter: Y

   The LOC expands the macros and compiles the expanded file to produce a JEDEC programming file (COMCODE.JED), a utilization report file (COMCODE.RPT), a minimized logic equation file (COMCODE.LEF) and an error message file (COMCODE.ERR). For traceability, a file called COMCODE.SDF is created to show the expanded form of the ADF output by the Macro Expander.

5. The LOC terminates execution with the following message:

   LOC cycle successfully completed

You can examine the LEF file to see the minimized form of the design. The LEF shows the EPLD primitives used to implement the design. Macro calls are not shown in the LEF. If you wish, you can also use LPS (Logic Programmer Software) to program a part.

September 1990

4

Getting Started with iPLS II/APT

DANIEL E. SMITH
PROGRAMMABLE LOGIC APPLICATIONS
INTEL CORPORATION

Order Number: 292067-001

GETTING STARTED WITH iPLS II/APT

CONTENTS

INTRODUCTION

This application note gets you up and running with iPLS II (Intel's Programmable Logic Software) on the PC platform in a short time. Installation instructions are provided in the *iPLS II User's Guide* (order number 450196). Refer to the main body of the *User's Guide* for detailed information on using iPLS II.

This session uses an ASCII text editor to create a netlist file called an ADF (Advanced Design File). Figure 1 shows the schematic diagram for the design used here. The sample circuit inputs a 4-bit value on IO0–IO3, increments it, and stores the new value in registers on the rising edge of CLK. This new value can then be output on IO0–IO3 by OE. The registers can be cleared by RESET or CLEAR. Output buffers can be three-stated by OE or RESET. A minute studying Figure 1 and a few minutes skimming the design primitives in Appendix A of this note is the only preparation you need for the design entry and compilation parts of this session. (Programming hardware and a blank 5AC312 device are required to complete the last part of the session.)

The ADF is created by a text editor (text editor supplied by user). The ADF contains several sections that define design information, inputs and outputs, target device, options, network connects, and logic equations. The completed ADF is translated into a JEDEC file by the LOC (Logic Optimizing Compiler). The JEDEC file can then be used by APT (Advanced Programming Tool) to program devices.

ENTERING A DESIGN

Step 1: Creating the ADF Header

Use your text editor to open an ASCII file named SAMP1.ADF. Enter the text shown here to create the Header Section of the ADF. The information in this section provides information to document the design. The header corresponds to title block information in schematic diagrams.

```
YOUR NAME
YOUR COMPANY
DATE
U1
REV. A
5AC312
Read 4 Bits, Increment, Store
PACKAGE:  D5AC312-25
```

Step 2: Creating the Declarations Section

Enter the following lines to create the Declarations Section of the ADF. The information in this section lists the target EPLD, specifies the setting of the Turbo Bit in the device, and defines all input and output pins.

```
OPTIONS:  TURBO = ON
PART:  5AC312
INPUTS:  OE, CLOCK, CLR, RESET
OUTPUTS:  IO0, IO1, IO2, IO3
```

Step 3: NETWORK: Section (Input Primitives)

Create the Network Section of the ADF. Network information defines the connections between inputs, macrocells, feedbacks, and logic equations in the ADF.

Type the keyword NETWORK: to start the Network Section, then list the inputs, as follows:

```
NETWORK:

OE = INP (OE)
CLOCK = INP (CLOCK)
CLR = INP (CLR)
RESET = INP (RESET)
```

The name on the left side of the equal sign (=) is the internal node name for the input signal. This internal node name must be used throughout the remainder of the Network and Equations Sections to make the connections to the respective input.

The name in parentheses on the right side is the external pin name and must match the corresponding name in the INPUTS: declaration. The internal node name (left side) can be the same as the pin name, or can be different.

Step 4: NETWORK: Section (Output Primitives)

The schematic in Figure 1 shows four registered outputs (with pins feeding back to the logic array). Include these primitives in the ADF as follows:

```
IO0,IO0f = ROIF(DOd,CLOCK,CLEARe,GND,OENe)
IO1,IO1f = ROIF(D1d,CLOCK,CLEARe,GND,OENe)
IO2,IO2f = ROIF(D2d,CLOCK,CLEARe,GND,OENe)
IO3,IO3f = ROIF(D3d,CLOCK,CLEARe,GND,OENe)
```

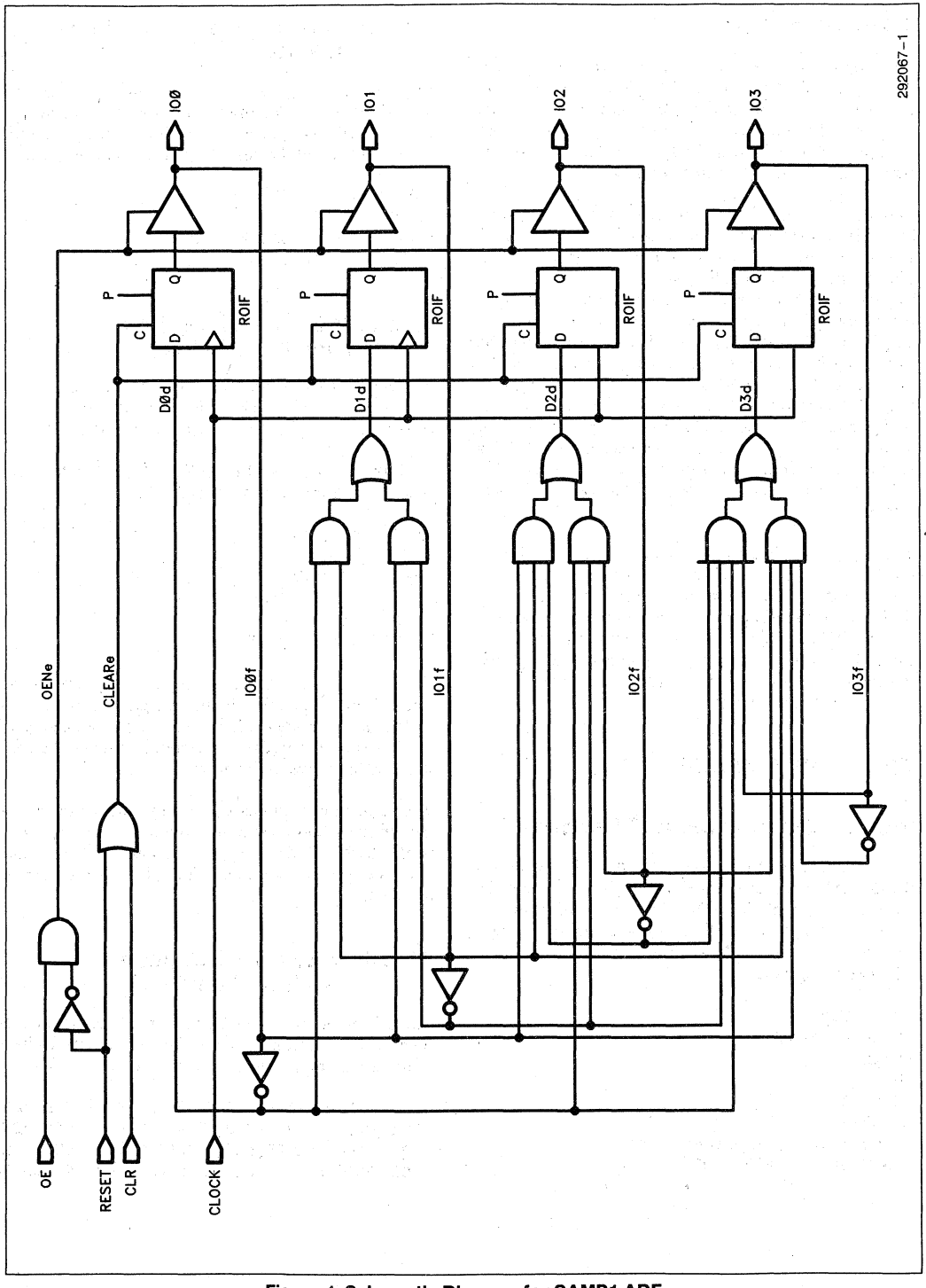
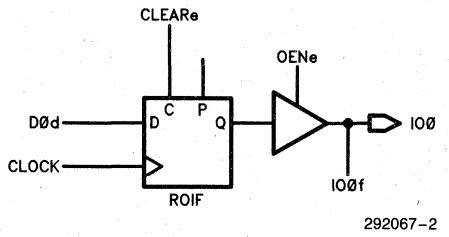


Figure 1. Schematic Diagram for SAMP1.ADF

The name on the left margin is the external pin name and must match the name in the OUTPUTS: declaration. The second name (after the comma) is the internal name of the feedback signal. This name can be the same as the output pin name, or it can be different. In this case, the same name is used with a lower case "f" to identify it as a feedback signal.

Output pin name and feedback names are followed by an equal sign (=) and the output primitive name. The output primitive name for all four entries is ROIF (Registered Output—Input Feedback). The node names in the parentheses define the input and control signal connections for the primitive.

Figure 2 shows a schematic representation of a ROIF primitive with the node names for the first ROIF entry. The input to the register is the node D0. The lower case "d" is a recommended convention used to identify the node as the input to a D-type register. The clock input is driven by input node CLOCK. Node CLEARe drives Clear, and node OENe drives Output Enable. The second ROIF statement uses a different node as the input to the register, but the clock and control lines are the same. No node name is provided for Preset; in this case the LOC will use the default (GND or "always disabled").



ADF EXAMPLE:
OUTPUTS: . . . , IO0, . . .
NETWORK:
IO0,IO0f = ROIF(D0d,CLOCK,CLEARe,GND,OENe)
EQUATIONS:
D0d = (Boolean Equation);

Figure 2. ROIF Primitive

The Preset (P), Clear (C), and Output Enable (Oe) control lines for all primitives can be driven by a node, explicitly tied high or low, or left to their default state.

Node names may be up to eight characters long. Oe (Output Enable), P (Preset), and C (Clear) may be connected to V_{CC} or GND. LOC recognizes V_{CC} and GND in both upper and lowercase letters. If no entry is made, C and P both default to GND (disabled); Oe defauts to V_{CC} (enabled).

Step 5: EQUATIONS: Section (Logic Equations)

The EQUATIONS: Section contains the logic equations for D0d, D1d, D2d, D3d, CLEARe, and OENe. Enter the EQUATIONS: keyword and the equations as follows:

```
EQUATIONS:

D0d = !IO0f;

D1d = !IO1f * IO0f
      + IO1f * !IO0f;

D2d = !IO2f * IO1f * IO0f
      + IO2f * !(IO0f * IO1f);

D3d = !IO3f * IO2f * IO1f * IO0f
      +  IO3f * !(IO2f * IO1f * IO0f);

OENe = OE * !RESET;

CLEARe = CLR + RESET;
```

These equations could have been implemented as a netlist in the NETWORK: section instead of the Equations Section. For example, a netlist implementation of D1d is as follows:

```
D1d = OR(D1d1, D1d2)
D1d1 = AND2(nIO1f, IO0f)
D1d2 = AND2(IO1f, nIO0f)
nIO1f = NOT(IO1f)
nIO0f = NOT(IO0f)
```

Step 6: Completing the ADF

Terminate the ADF with the following line:

```
END$
```

Save the file and exit the text editor program. Figure 3 shows the completed ADF.

```
YOUR NAME
YOUR COMPANY
DATE
U1
REV. A
5AC312
Read 4 Bits, Increment, Store
PACKAGE: D5AC312-25
OPTIONS: TURBO = ON
PART:    5AC312

INPUTS:   OE, CLOCK, CLR, RESET

OUTPUTS:  IO0, IO1, IO2, IO3

NETWORK:
          OE = INP (OE)
          CLOCK = INP (CLOCK)
          CLR = INP (CLR)
          RESET = INP (RESET)
          IO0, IO0f = ROIF (DOd, CLOCK, CLEARe, GND, OENe)
          IO1, IO1f = ROIF (D1d, CLOCK, CLEARe, GND, OENe)
          IO2, IO2f = ROIF (D2d, CLOCK, CLEARe, GND, OENe)
          IO3, IO3f = ROIF (D3d, CLOCK, CLEARe, GND, OENe)

EQUATIONS:
          DOd = !IO0f;

          D1d = !IO1f * IO0f
              + IO1f * !IO0f;

          D2d = !IO2f * IO1f * IO0f
              + IO2f * !(IO1f * IO0f);

          D3d = !IO3f * IO2f * IO1f * IO0f;
              + IO3f * !(IO2f * IO1f * IO0f);

          OENe = OE * !RESET;

          CLEARe = CLR + RESET;

END$
```

Figure 3. Listing for SAMP1.ADF

COMPILING THE DESIGN (LOC)

Submit the sample design to the LOC by entering the following command line:

```
LOC SAMP1 <Enter>
```

The LOC executes, generating SAMP1.RPT (utilization report file) and SAMP1.JED (JEDEC programming file). A minimized logic equation file (LEF) and an error file can also be generated via command line options. Refer to Chapter 3 of the *iPLS II User's Guide* for a discussion of these options.

As the LOC executes, it displays the following messages:

```
***INFO-XLT-Read 4 Bits, Increment, Store
***INFO-LOC-ADF converted to LEF; sampl
***INFO-LOC-Sum Of Products (S.O.P) LEF produced
***INFO-Espresso II-mv (C) Copr. 1985 U.C. Berkeley
***INFO-LOC-LEF reduced
***INFO-LOC-Resource demand determined
***INFO-FIT-DeMorgan's inversion of CLEAR signal CLEARe on IO0
***INFO-LOC-Design fitting complete
***INFO-LOC-JEDEC file output

LOC cycle successfully completed
```

LOC returns control to DOS when it completes.

PROGRAMMING A DEVICE (APT)

The following programming hardware is required to work through this part of the session:

- iUP-PC Universal Programmer-Personal Computer or iUP-200A/201A Universal Programmer
- iUP-GUPI Module base
- GUPI LOGIC-IID Programming Adaptor
- Blank 5AC312 EPLD

Proceed as follows:

Step 1: Invoking APT

Invoke APT the command line as follows:

```
APT <Enter>

APT Release [x.y] SID [info.]
Copyright Intel Corporation, 1989
Welcome to APT

File 'apt.cfg' Does Not Exist, Create? Y/N [Y|y]: <Enter>

APT>>h
```

The help command provides a quick reference to APT commands.
Usage:

```
?|h[elp][command] where command is one of the following:

      b[lankcheck]
      c[hecksum]
      d[efaults]
      r[ead]
      p[rogram]
      v[erify]
      q[uit]
      '\' to Escape
```

The prompt for APT is as follows:

```
APT>>
```

Step 2: Checking for a Blank Device

Insert the 5AC312 into the 24-pin DIP socket on the LOGIC-IID Adaptor. Secure the device with the lever. Enter the BLANKCHECK command as shown below (the -D and device name are needed to identify the device). Note that since this is the first time that the programmer is accessed since invocation, APT initializes the programmer before executing the blankcheck. Also, since no device-specific software (.DSS) files have previously been loaded, the 5AC312.DSS file is loaded.

```
BLANKCHECK -D 5AC312 <Enter>

Initializing Serial Port COM3_PCPP
Initializing Device 5ac312
Initializing PCPP
PCPP Version: PCPP EPROM Loader V1.0, 08-05-86, 14:25
Copyright Intel Corporation, 1986
Running Diagnostics on PCPP
Downloading 'c:\iplsii\pcpp85.obj'
Downloading 'c:\iplsii\5ac312.dss'
Building Physical Data Base for 5ac312

Insert the 5ac312 device and press return to continue or '\' to Escape:
<Enter>
          Device is Blank

Insert the 5ac312 device and press return to continue or '\' to Escape: \
<Enter>
```

Use a backslash (\) followed by <Enter> to exit the blankcheck loop.

Step 3: Programming the Device

Program the device using SAMP1.JED and using the defaults for all other options;

```
PROGRAM SAMP1.JED<Enter>

Reading JEDEC Fuse Data from 'SAMP1.jed' at address XXXX
Computed checksum = YYYYH, file checksum = ZZZZH
Finished Reading JEDEC file 'SAMP1.jed'.
  Turbo is ON
  Verify Protect is OFF

Insert the 5AC312 device and press return to continue or '\' to Escape: \
<Enter>
```

Use a backslash (\) followed by <Enter> to exit the program loop.

Execute the QUIT command to exit to the OS/command shell:

```
QUIT<Enter>
EXITING APT
```

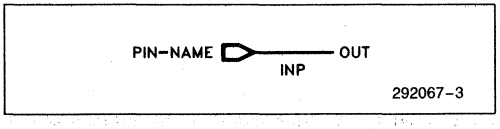
This completes the session for iPLS II. Refer to the *User's Guide* for additional information on the different aspects of EPLD design using iPLS II. Refer to the *APT User's Guide* for information on programming with APT.

APPENDIX A

This appendix shows the INP and ROIF design primitives used in the sample session with the 5AC312. The function and syntax for each primitive is provided and a schematic symbol of the primitive is shown.

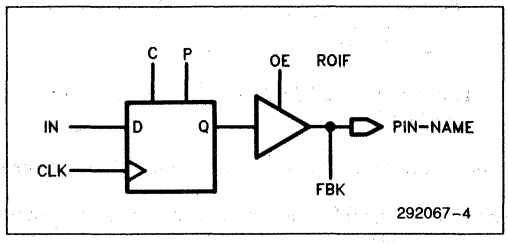
Table B-1 in the *iPLS II User's Guide* lists all primitives valid for all Intel EPLDs.

INP



ADF Name: INP (Primary Input)
ADF Syntax: Out = INP(In)
Schematic Name: INP (Primary Input)
Description: Outputs:
Out = output to logic array
Inputs:
In = pin input

ROIF



ADF Name: ROIF (Registered Output, Input Feedback)
ADF Syntax: Out, Fbk = ROIF(In,Clk,C,P,Oe)
Schematic Name: ROIF (Registered Output, Input Feedback)
Description: Outputs:
Out = output to EPLD pin
Fbk = feedback to logic array
Inputs:
In = input from logic array
Clk = register clock
C = clear input
P = preset input
O_e = 3-state buffer output enable input

September 1990

4

Getting Started with IPLDview-286

DAVE PISTONE
DANIEL E. SMITH
PROGRAMMABLE LOGIC APPLICATIONS
INTEL CORPORATION

Order Number: 292068-002

GETTING STARTED WITH IPLDVIEW-286

CONTENTS

INTRODUCTION

This IPLDview-286 tutorial provides a step-by-step procedure for a typical EPLD design cycle, from schematic capture to simulation. The steps are summarized as follows:

1. Enter the design via the IPLDdraw schematic interface.
2. Simulate the design via IPLDsim before fitting it into a device (precompilation simulation).
3. Netlist the design to an ADF when satisfied with the results.
4. Compile the design using the iPLS II software.
5. Translate the JEDEC file back into a simulation file, with device timing.
6. Simulate the design via IPLDsim with the device parameters (post-compilation simulation).

A typical design flow involves multiple iterations of some steps to achieve the desired results. Figure 1 shows the relationship of drawing, compilation, and simulation. If you are using IPLDview-286, the design flow includes all seven steps. If you are only using IPLDsim, the design flow starts at step 4. If you are only using IPLDdraw, the design flows includes steps 1, 3 and 4 only.

NOTES:

This tutorial concentrates on IPLDview-286 features that are different from the standard VIEW*logic** schematic capture and simulation products (i.e., features unique to IPLDview-286, IPLDdraw or IPLDsim). For a description or explanation of how to use standard features (panning, zooming, etc.), refer to the "Digital Tutorial" section of the Workview* Manual. Compilation and programming are covered in a related application note, AP-332, *Getting Started with iPLS II/APT*, order number: 292067.

ENTERING A SCHEMATIC

This first section is a step-by-step tutorial for entering an EPLD design in IPLDdraw or IPLDview-286. References are made to the menu selections. All of the menu selections have a keyboard entry equivalent. These are listed in the Workview manuals. Before starting IPLDdraw or IPLDview-286, the software must be installed and the appropriate security block must be connected to the mouse port on your system. If this has not been done already, refer to the installation guide(s).

1. Enter Workview from DOS by typing:

   WV <Enter>

2. Open a schematic window by making the following menu selections:

   Window Open Viewdraw Schematic

3. Enter the name of the design:

   COUNTER <Enter>

4. You are prompted to specify an area to open the window. Use the left mouse button to specify the area by clicking once for one corner, moving the mouse to the other corner, and clicking again.

NOTE:

Double clicking in the same place causes the window to take all available screen space.

5. Create a component by making the following menu selection:

   Create Component

6. Use the middle mouse button to execute the command, then specify the name of the component (symbol) to be created.

   RORF <Enter>

7. Move the component by moving the mouse. Place it by clicking the middle button on the mouse.

*Workview and VIEW*logic* are registered trademarks of Viewlogic Systems, Inc.

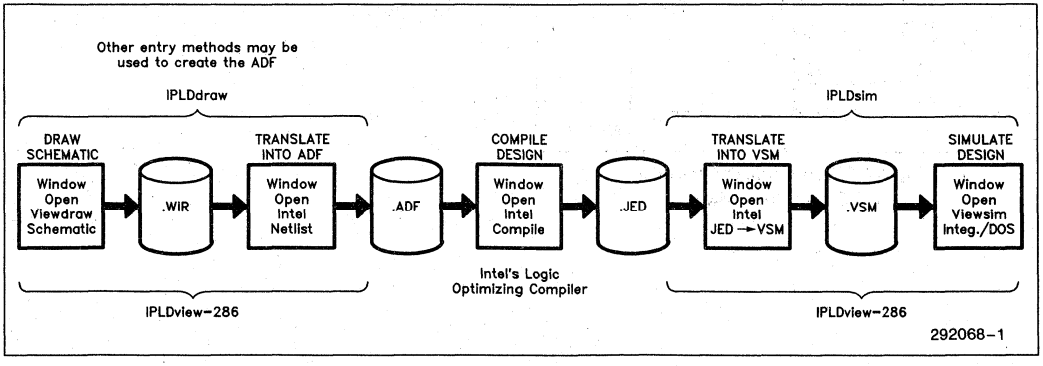


Figure 1. EPLD Design Flow with IPLDView-286

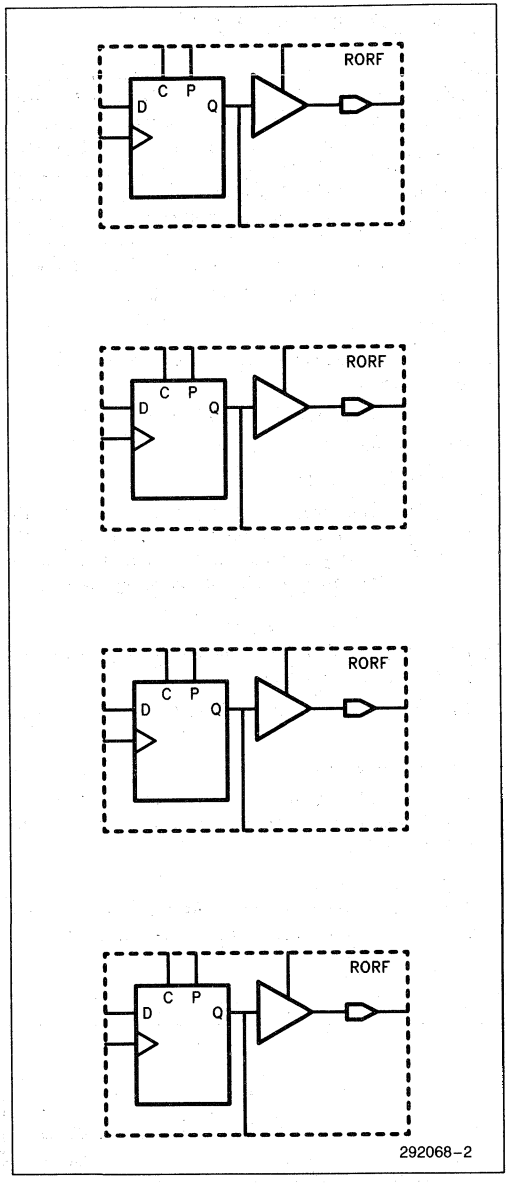


Figure 2. Four RORFs Placed on Schematic Sheet

8. Repeat the last two steps for the remainder of the components in Figure 3 (HEADER, INP, NOT, AND2, AND3, AND4, OR2, OR3, OR4, CLKB, IN & OUT).

NOTE:
Try using the MOVE and COPY menu commands also. To delete a component, select it with the left button and then use the menu to select the DELETE function.

9. The design should now resemble Figure 3.
10. To add wire (NETS), make the following menu selection:

    Create Net

11. Position the cursor at a component pin and click the middle mouse button. Then move the mouse to draw the net. To bend the net, click the middle mouse button. To end the net at another component pin, position the cursor on it and click the middle button. To end the net in free space, click the middle button and then the right button.
12. Repeat the last step until all the nets in Figure 4 have been drawn. The design should now resemble Figure 4.
13. To label a net, make the following menu selection:

    Create Label

14. Use the left button to select the net to be labeled. Press the middle button to execute the command. You are prompted for the net name. Enter the name and press <Enter>, as follows:

    Enter the name, followed by a <Enter>.

    A <Enter>

    Position the text string with the mouse and place it by clicking the middle mouse button. The label is attached to the selected net, regardless of where it is placed on the drawing.

15. Repeat step 14 for the remainder of the nets to be labeled in Figure 4. The design should now resemble Figure 5.

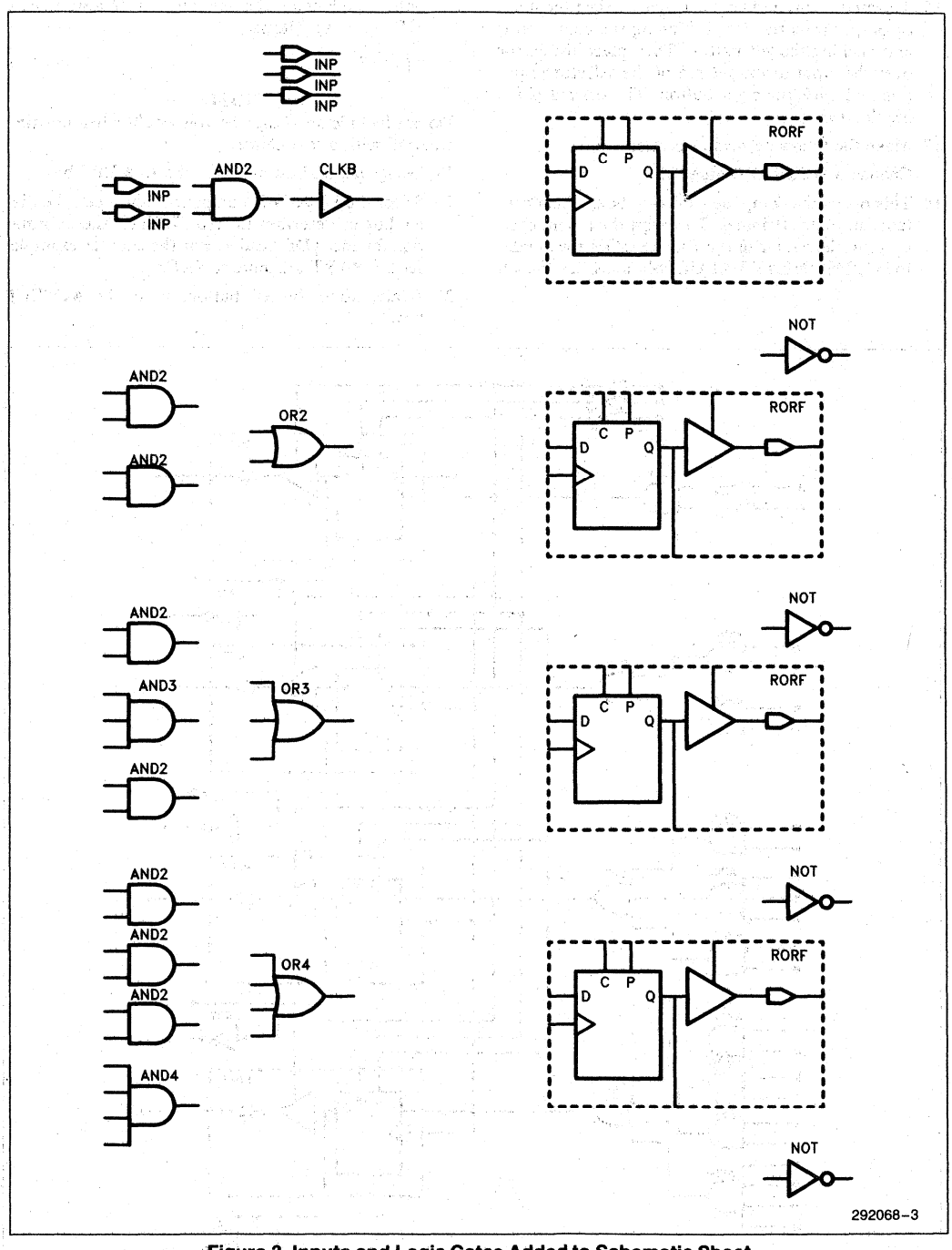


Figure 3. Inputs and Logic Gates Added to Schematic Sheet

16. To assign a pin number to a signal, select the input or output primitive by positioning the cursor on it and clicking the left button. Then place the cursor over the input or output pin of the selected primitive and click the right button. The desired pin is highlighted.
17. Make the following menu selection:

    Change Attr(ibutes) Dialog All
18. This opens the components dialog box, which contains all of its attributes. To assign the pin number, position the cursor in the PINASSIGN row, under the COMPONENT VALUE column. Click the left button and enter the pin number followed by a <Enter>, as follows:

    20 <Enter>

NOTE:
Do not include an @ sign as you would when creating an ADF with a text editor.

19. Using the left button, select the ACCEPT box.
20. Select the HEADER component and open the dialog box as described in step 17. Enter the information for the ADF header. For the counter example, set the PART attribute to 5AC312.
21. Again, using the left button, select the ACCEPT box.

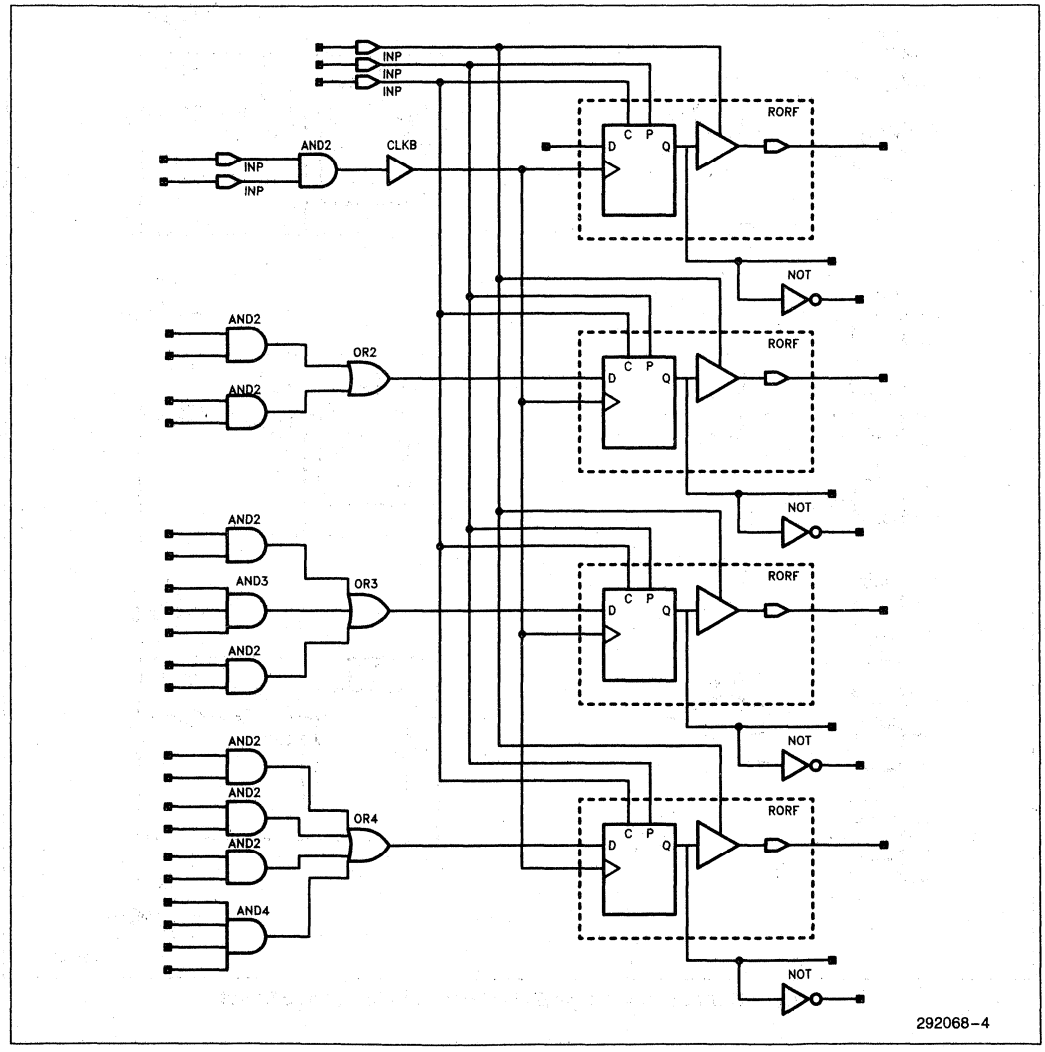


Figure 4. Wires (NETS) Added to Schematic Sheet

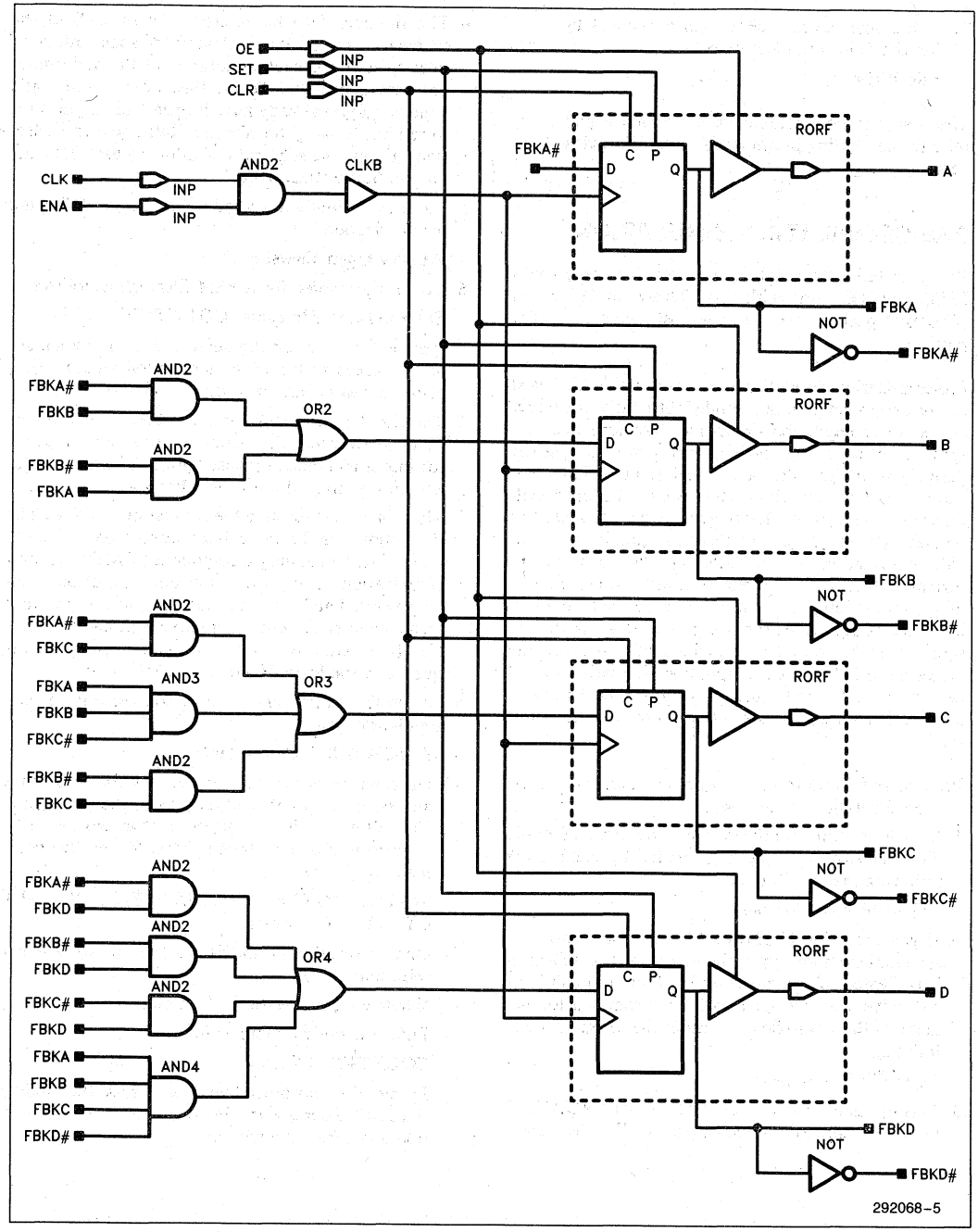


Figure 5. Completed 4-Bit Counter Schematic

22. The design can now be written to the disk by making the following menu selection:

    File Write

The design is now completed. To obtain a plot of it, refer to the plotting information in the Workview manuals.

PRE-COMPILATION SIMULATION

Pre-compilation simulation is provided only with IPLDview-286, not with IPLDdraw or IPLDsim (IPLDsim package does provide post-compilation support).

Pre-compilation simulation uses a simulation file based on the design before it is compiled/fitted into the target device. Pre-compilation simulation, therefore, does not provide device timing and does not guarantee that the design will fit into the target device. Pre-compilation simulation, however, allows the design to be functionally verified early in the design cycle, where changes can be quickly made. Post-compilation simulation can provide timing information of fitted designs when this information is needed. The recommended procedure is to use pre-compilation simulation to verify the functionality of the design. Once the design is functional, the design should be compiled to make sure it fits into the target device, then post-compilation simulation should be performed to verify timing. (Compilation and post-compilation simulation are discussed later in this tutorial.)

Proceed as follows to perform pre-compilation simulation on the 4-bit counter design:

1. In order to simulate the design, a simulation (.VSM) file is needed. This file is generated by making the following menu selection:

   Export Wirelist Viewsim

2. If you are using IPLDview-286 and have an active schematic window open, the netlist for that schematic is created. If you are using IPLDsim or don't have an active window open, you are prompted for the name of the simulation file. Enter the design name as follows:

   COUNTER <Enter>

3. This opens a DOS window and runs the VIEWSIM netlister, generating a file called COUNTER.VSM.

4. The simulator can be entered with or without the schematic window open. Having the schematic window open allows back annotation of the node states onto the schematic. To do this, open a schematic window (if you already have it open, you might want to reframe it using the Window Reframe menu selection). Frame the schematic window to use about the bottom half of the screen.

5. To open a simulation window, make the following menu selection:

   Window Open Viewsim

6. You are prompted for a VSM filename as follows:

   Enter network file name [COUNTER]:

   Press enter to accept the default. Use the remainder of the screen to frame the simulation window (i.e., click the center button to fill the screen).

7. The simulation window is now the active window. To make the schematic window active, move the cursor into it and click a mouse button. The node states are back annotated onto the schematic.

8. Move back and reactive the simulation window. The simulation can be done interactively on the command line, by running a command (.CMD) file, or a combination of the two. Input and output data can be generated and viewed in tabular form or graphics form. This session uses interactive command line entry. (Appendix A shows the same command sequence in the form of a command file.)

9. Assign the output signals to be tracked during each simulation step:

   WATCH A B C D <Enter>

10. Open an output stream to the waveform software and assign all inputs and outputs to the stream file. This allows all input and output waveforms to be displayed after simulation steps. Enter the command as follows:

    WAVE COUNTER.WFM CLK ENA CLR SET OE A B C D <Enter>

11. Open a waveform window via the following menu selection:

    Window Open Viewwave Viewsim

12. Enter the design name as follows:

    COUNTER <Enter>

    Frame the waveform window across the middle third of the screen. It will cover part of the simulation and schematic windows.

13. Select the simulation window and enter commands to place all inputs to a known state. For this session place CLK, CLR, and SET low and ENA and OE high. Then run one simulation cycle:

    L CLK CLR SET <Enter>
    C <Enter>
    H ENA OE <Enter>
    C <Enter>

14. Reset the counter by cycling CLR high, then low:

    H CLR <Enter>
    C <Enter>
    L CLR <Enter>
    C <Enter>

15. Move the mouse to the waveform window and click on it to update the waveforms. Your screen should appear as shown in Figure 6.

16. Click on the simulation window and define the clock signal as a high followed by a low. Run 8 simulation cycles:

    CLOCK CLK 1 0 <Enter>
    C 8 <Enter>

17. Run 8 additional simulation cycles, then disable the clock and test the SET signal:

    C 8 <Enter>
    L ENA <Enter>
    C 2 <Enter>
    H ENA <Enter>
    C 2 <Enter>
    H SET <Enter>
    C <Enter>
    L SET <Enter>
    C <Enter>

18. Move the mouse to the waveform window and click on it to update the waveforms. Your screen should appear as shown in Figure 7.

19. To save the waveform window for later viewing, make the following menu selection:

    File Write Savefile

20. Close the simulation window by clicking on it and selecting:

    Window Close

    Close the waveform window in the same way.

This completes the pre-compilation simulation session. The entire session can also be executed automatically using the command file shown in Appendix A.

4

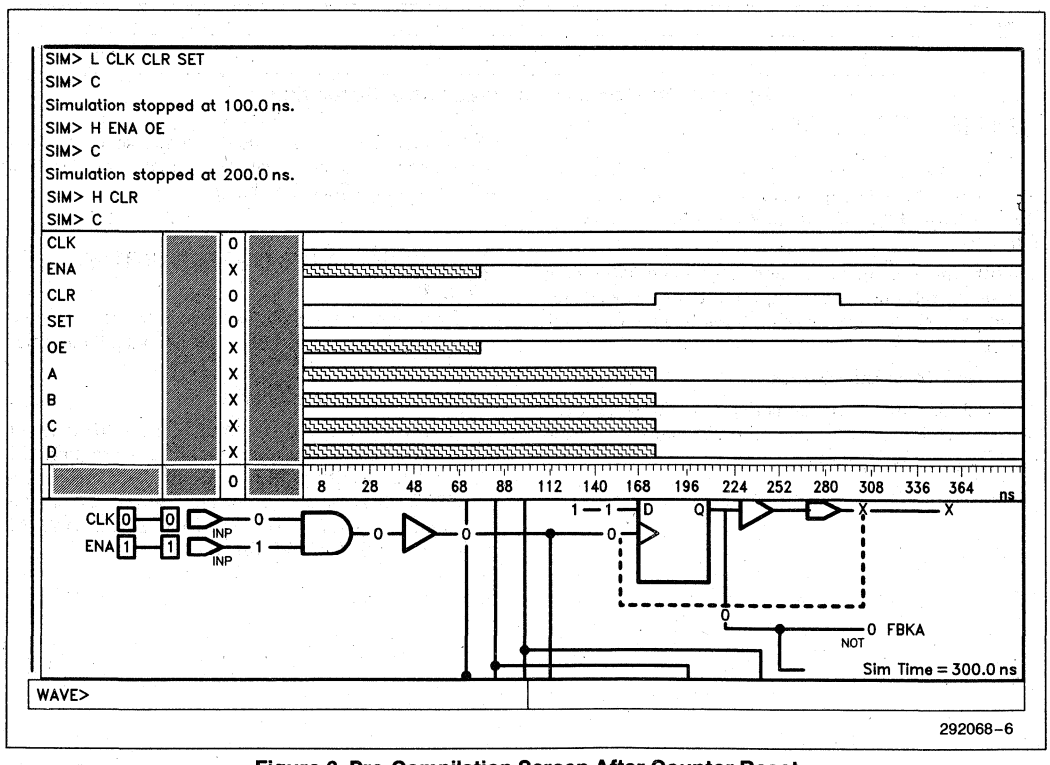


Figure 6. Pre-Compilation Screen After Counter Reset

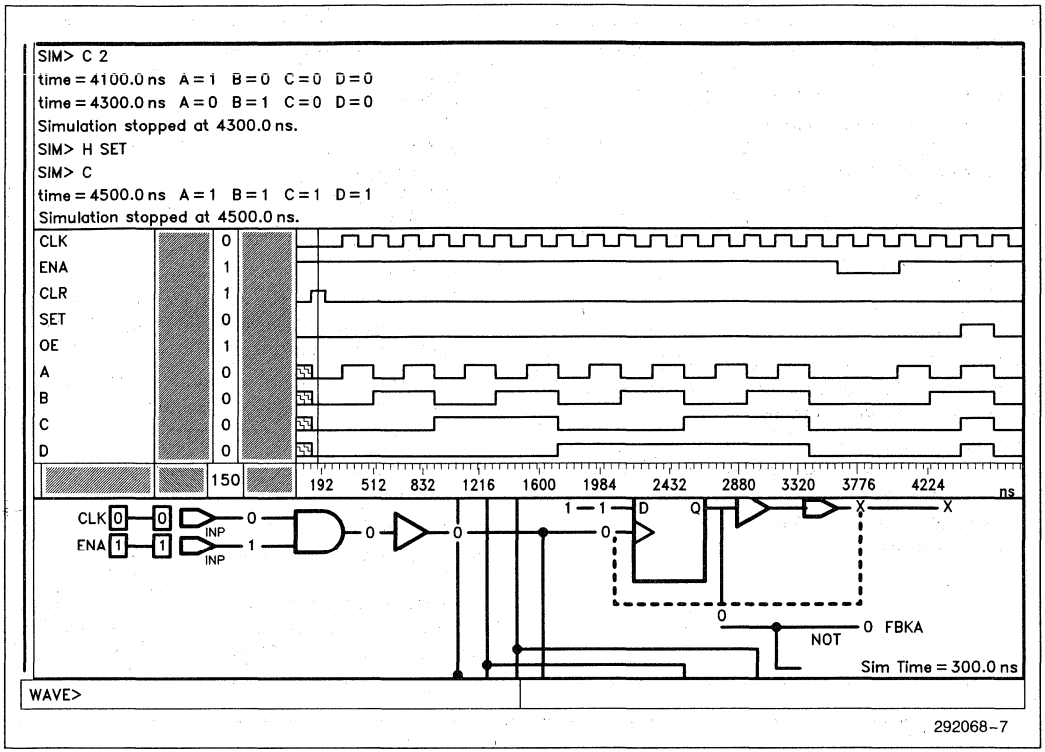


Figure 7. Pre-Compilation Screen After Simulation

NETLISING TO ADF

Once the design is functional, it can be netlisted to ADF (Advanced Design File) format for use by the iPLS II LOC (Logic Optimizing Compiler). Proceed as follows:

1. With the schematic window still open (you may want to reframe it to occupy the entire screen), invoke the Netlister by making the following menu selection:

   Window Open Intel Netlist

2. The design that is in the active window is netlisted into ADF format. The file is stored in the \USER\SCH directory.

NOTE:
If any modifications are made to the schematic and the schematic is not written to disk, the resulting ADF will not reflect the changes.

COMPILING THE DESIGN WITH iPLS II

1. After the design has netlisted successfully, the compiler can be run by making the following menu selection:

   Window Open Intel Compile

2. This invokes the LOC to compile the ADF to generate output files, including the JEDEC programming file. The resulting files will also be put in the \USER\SCH directory. To modify the way that the compiler is invoked, edit the INTELCMP.BAT file in the \WORKVIEW directory.

TRANSLATING THE DESIGN INTO POST-COMPILATION SIMULATION FILE

If post-compilation simulation is desired, the design must be translated from a JEDEC file into a .VSM file. Proceed as follows:

1. Invoke the JEDEC Translator by making the following menu selection:

   Window Open Intel JED = > VSM

2. This opens a DOS window and runs the Translator. You are prompted for the EPLD base name, timing file, package type, and the JEDEC file name. For the counter example, the prompts and responses are as follow:

   A) Enter base pld model name: 5AC312 <Enter>

   B) Enter device speed bin: 5AC31225 <Enter>

   Enter 5AC31200 for zero delay device model.
   Enter 5AC31225 for the -25 device model.
   Enter 5AC31230 for the -30 device model.
   Enter 5AC31235 for the -35 device model.

   C) Enter device package type: D5AC312 <Enter>

   Enter D5AC312 for the DIP package.
   Enter N5AC312 for the PLCC package.

   D) Enter JEDEC file name (no extension): COUNTER <Enter>

3. The PLD model, the package and timing files and the JEDEC file are read and the translator generates the .VSM file. This process takes approximately 5 minutes on a 16 MHz 80386-based PC.

POST-COMPILATION SIMULATION

Post-compilation simulation is provided in the IPLDview-286 and IPLDsim packages, not in the IPLDdraw package.

Post-compilation simulation differs from the pre-compilation simulation in that post-compilation provides device timing, and is simulating a design after it has been fitted into a target device. Post-compilation simulation allows selection of device package and speed bin.

The procedure for doing post-compilation is similar to that for pre-compilation simulation. A .VSM file is required, and must be created from the JEDEC programming file for the design. Generation of the .VSM for post-compilation simulation takes longer and requires more memory than for pre-compilation simulation. One reason for this is the fact that post-compilation simulation uses the actual device model as a base. A limitation of post-compilation simulation (in comparison to pre-compilation simulation) is that back annotation of node states onto schematics is not supported. Because the simulation data base is different than the schematic data base, unknown nodes will be flagged. This can be solved by not returning to the schematic data base while the simulation window is open, or by turning the display of node values off, as follows:

Change Display Values Off

Another important consideration with post-compilation simulation is emulating the power-on reset feature of the actual device. (Macrocell registers in Intel EPLDs power up low.) The simulator does provide this feature internally. In order to emulate it, there is a special signal called INTERNAL_PORS that is used to provide the power on reset. This signal name is reserved, and cannot be used as a signal name in your design.

1. To open a simulation window, make the following menu selection:

   Window Open Viewsim

   Frame the window on the bottom two-thirds of the screen. The simulation can be done interactively on the command line, by running a command (.CMD) file, or a combination of the two. Input and output data can be generated and viewed in tabular form or graphics form. This session uses interactive command line entry. (Appendix A shows the same command sequence in the form of a command file.)

2. Assign the output signals to be tracked during each simulation step:

   WATCH A B C D <Enter>

3. Open an output stream to the waveform software and assign all inputs and outputs to the stream file. This allows all input and output waveforms to be displayed after simulation steps. Enter the command as follows:

   WAVE COUNTER.WFM CLK ENA CLR SET OE A B C D <Enter>

4. Open a waveform window via the following menu selection:

   Window Open Viewwave Viewsim

5. Enter the output file stream name as follows:

   COUNTER <Enter>

   Frame the waveform window across the top third of the screen.

6. Enter commands to place all inputs to a known state. For this session place CLK, CLR, and SET low and ENA and OE high. Then run one simulation cycle:

   L CLK CLR SET <Enter>
   H ENA OE <Enter>
   C <Enter>

7. Reset the registers by cycling INTERNAL_PORS high, then low:

   H INTERNAL_PORS <Enter>
   C <Enter>
   L INTERNAL_PORS <Enter>
   C <Enter>

   The above sequence can also be incorporated at the beginning of the command file to be executed. If power-on reset is not desired, then the INTERNAL_PORS signal must be held low before simulating. This can be done as follows:

   L INTERNAL_PORS <Enter>
   CYCLE <Enter>

   If INTERNAL_PORS is left in the unknown state, all registered outputs will also remain in the unknown state throughout simulation.

8. Define the clock signal as a high followed by a low and run 8 simulation cycles:

   CLOCK CLK 1 0 <Enter>
   C 8 <Enter>

9. Move the mouse to the waveform window and click on it to update the waveforms. Your screen should appear as shown in Figure 8.

10. Run 8 additional simulation cycles, then disable the clock and test the SET and CLR signals:

    C 8 <Enter>
    L ENA <Enter>
    C 2 <Enter>
    H SET <Enter>
    C <Enter>
    L SET <Enter>
    C <Enter>
    H CLR <Enter>
    C <Enter>
    L CLR <Enter>
    C <Enter>

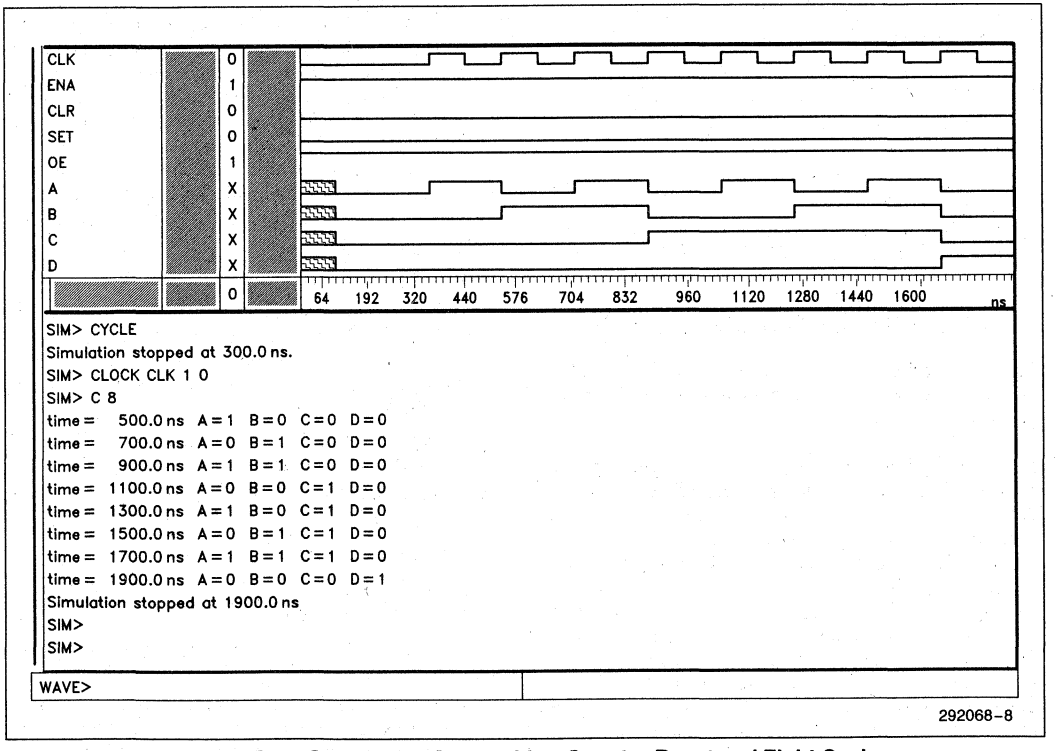


Figure 8. Post-Compilation Screen After Counter Reset and Eight Cycles

11. Move the mouse to the waveform window and click on it to update the waveforms. Your screen should appear as shown in Figure 9.
12. Move the mouse to the rising edge of CLK at the 1300 ns point and click the middle button. A vertical timeline is displayed across all signals at that point. The lower right status window displays the following:

    T = 1300 DT = 1300

    The T indicates the time from the start of the simulation, 1300 ns in this case. The D indicates the delta from the previous timeline, 1300 in this case since this is the first timeline.
13. Move the mouse button to the first transition on output A after the 1300 ns clock edge and click the middle button. A timeline is displayed and the status window is updated as follows:

    T = 1325 DT = 25

    The new time from simulation is shown, along with a 25 ns delay from the first timeline. This delay corresponds to the asynchronous clock-to-output spec. for the 5AC312-25 device.
14. To save the waveform window for later viewing, make the following menu selection:

    File Write Savefile
15. Close the simulation window by clicking on it and selecting:

    Window Close

    Close the waveform window in the same way:

This completes the post-compilation simulation session. The entire session can also be executed automatically using the command file shown in Appendix A.

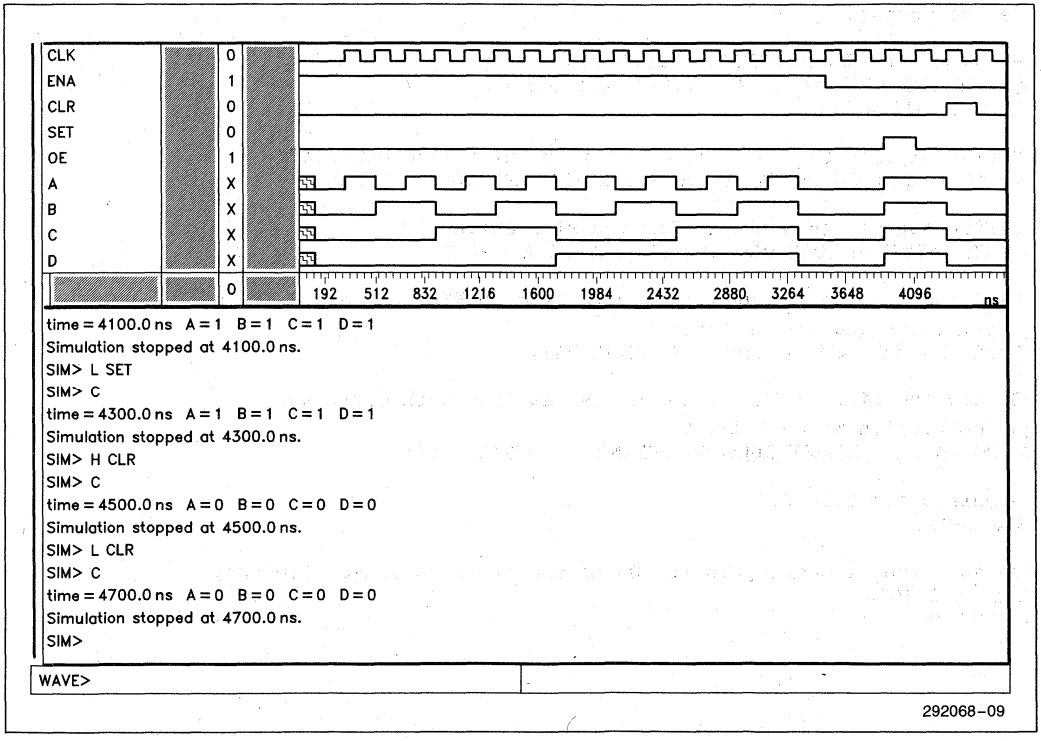


Figure 9. Post-Compilation Screen After Simulation

APPENDIX A

This appendix includes a command file, input vector file, and output file for simulating the example counter design (number and order of cycles is different). Refer to the Viewsim manual for additional information on use of command/vector files.

Command File (COUNTER.CMD)

The command filed controls simulation for the example design. The node INTERNAL__PORS should only be included for post-compliation simulation. To run the command file, open a simulation window with the network name COUNTER. After the COUNTER.VSM file is loaded, type COUNTER.CMD <Enter> at the simulation command prompt.

```
|Command file for the counter example
|Define vectors

|Note: use INTERNAL_PORS for post-comp. only
VECTOR INS INTERNAL_PORS CLK ENA CLR SET OE
VECTOR OUTS A B C D

|Define signals to store graphically in the file COUNTER.WFM
WAVE COUNTER.WFM INTERNAL_PORS CLK ENA CLR SET OE A B C D

|Define signals to watch during the simulation
WATCH INTERNAL_PORS CLK ENA CLR SET OE A B C D

|Read and apply a new vector of data from COUNTER.PAT every 100 ns.
|Apply it to the vector "INS"
EVERY 100 DO (ASSIGN INS < COUNTER.PAT)

|Sample the pins on the watch list 90 ns into each cycle and
|print results to COUNTER.OUT
AFTER 900 DO (EVERY 1000 DO (PRINT > COUNTER.OUT))

|Simulate for 6400 ns
SIM 64000

|Or set size of each cycle to 100 ns and cycle 32 times (6400 ns)
STEPSIZE 1000
|CYCLE 32
```

Vector File (COUNTER.PAT)

This file provides the input vectors specified in the command file.

```
|Test vectors for counter example
|
|
|CECS
|LNLEO
|KARTE
|-----
00100
00100
01001
11001
01001
11001
01001
11001
01001
11001
01001
11001
01001
11001
01001
11001
01001
11001
01001
11001
01001
11001
01001
11001
01001
11001
01001
11001
01001
11001
01001
11001
01001
11001
01000
11000
01000
11000
01000
11001
```

VECTOR FILE (COUNTER.PAT) (Continued)

```
01001
11001
01001
11001
|more input vectors
01001
11001
01001
11001
01001
11001
01001
11001
01001
11011
01011
11011
01011
11001
01001
10001
00101
10101
01101
11101
```

Output File (COUNTER.OUT)

This file shows the results of simulation.

```
          ICECSOABCD
          NLNLEE
          TKART
          E
          R
          N
          A
          L
          -
          P
          O
          R
          S
          ----------
     TIME BBBBBBBBBB

  90.0 ns 000100ZZZZ
 190.0 ns 000100ZZZZ
 290.0 ns 0010010000
 390.0 ns 0110011000
 490.0 ns 0010011000
 590.0 ns 0110010100
 690.0 ns 0010010100
 790.0 ns 0110011100
 890.0 ns 0010011100
 990.0 ns 0110010010
1090.0 ns 0010010010
1190.0 ns 0110011010
1290.0 ns 0010011010
1390.0 ns 0110010110
1490.0 ns 0010010110
1590.0 ns 0110011110
1690.0 ns 0010011110
1790.0 ns 0110010001
1890.0 ns 0010010001
1990.0 ns 0110011001
2090.0 ns 0010011001
2190.0 ns 0110010101
2290.0 ns 0010010101
2390.0 ns 0110011101
2490.0 ns 0010011101
2590.0 ns 0110010011
2690.0 ns 0010010011
2790.0 ns 0110011011
2890.0 ns 0010011011
2990.0 ns 0110010111
3090.0 ns 0010010111
3190.0 ns 0110011111
3290.0 ns 0010011111
3390.0 ns 0110010000
3490.0 ns 001000ZZZZ
3590.0 ns 011000ZZZZ
3690.0 ns 001000ZZZZ
3790.0 ns 011000ZZZZ
3890.0 ns 001000ZZZZ
```

Appendix 5

APPENDIX
THIRD-PARTY SUPPORT
(Support claimed by manufacturer)

PLD PROGRAMMING SUPPORT

Vendor	Product	Module	Adaptor/Device/Version/Etc.
Advin	Sailor PAL/SA	—	DIPs: 5C031, 5C032, 5C060
	Sailor PAL/SB	—	(same as above)
		w/EM-900	DIP: 5C090
		w/EM-1800	PLCC: 5C180
		w/PLCC Adap.	PLCCs: 5C060, 5C090
BP Micro	PLD-1100	—	V1.46—DIPS: 85C220, 85C224, 5AC312, 5C031, 5C032, 5C060 w/socket converter from outside source—PLCCs: 85C220, 85C224, 5AC312, 5C060
	PLD-1128	—	V1.46—DIPs: 85C220, 85C224, 5AC312, 5C031, 5C032, 5C060 w/socket converter from outside source—PLCCs: 85C220, 85C224, 5AC312, 5C060
Bytek	135H-U	UNICEL	w/LTA00C—DIPs: 5C031, 5C032, 5C060, 5C090, 5C180
	145H-AD	—	DIPs: 5C031, 5C032, 5C060, 5C090, 5C180
Data I/O	UNISITE	SITE 40/48	V3.1—DIPS: 85C220, 85C224, 85C060, 85C508, 5AC312, 5AC324, 5C031, 5C032, 5C060, 5C090
		CHIPSITE	V3.1—PLCCs: 85C220, 85C224, 85C060, 85C508, 5AC312, 5AC324, 5C060, 5C090, 5C180
	Model 2900	—	V1.1—DIPs: 85C220, 85C224, 85C060, 85C508, 5AC312, 5AC324, 5C031, 5C032, 5C060, 5C090 V1.1—PLCCs: 85C220, 85C224, 85C060, 85C508, 5AC312, 5AC324, 5C060, 5C090
	Model 29B	LogicPak V.4	303A-011A (V10)—DIPs: 85C220, 85C224, 85C060, 5AC312, 5C031, 5C032, 5C060 303A-011B (V05)—PLCCs: 85C220, 85C224, 85C060, 5AC312, 5C031, 5C032, 5C060 303A-010 (V03)—DIP & PLCC: 85C090, 5C090
	Model 60A/H	—	DIPs: 5C031, 5C032, 5C060
Digelec	860	—	DIPs: 5C031, 5C032, 5C060, 5C090
		w/8601	PLCCs: 5C060, 5C090, 5C180
Elan	1014	—	DIPs: 5C032, 5C060, 5C090 w/socket converter from outside source—PLCCs: 5C060, 5C090
	5-145	—	DIPs: 5C032, 5C060, 5C090 w/socket converter from outside source—PLCCs: 5C060, 5C090

5

PLD PROGRAMMING SUPPORT (Continued)

Vendor	Product	Module	Adaptor/Device/Version/Etc.
Intel	iUP-PC iUP-200A/201A	GUPI Base (both prog.)	GUPI LOGICIID—DIPs: 85C060, 85C090, 5AC312, 5C060, 5C090 w/ADAPT24TO28—PLCCs: 85C060, 5AC312, 5C060 w/ADAPT40TO44—85C090 (PLCC), 5C090 (PLCC) GUPI 20D20J—85C220 (DIP & PLCC), 5C032 (DIP), 5C031 (DIP) GUPI 24D28J—85C224 (DIP & PLCC) GUPI 40D44J—5AC324 (DIP & PLCC) GUPI LOGIC-18—5C18[illegible] (PLCC) GUPI LOGIC-18G—5C180 (PGA)
Kontron	EPP-80 MPP-80	UPM/B UPM/C UPM/B UPM/C	5C031, 5C032, 5C060, 5C090, 5C180 5C031, 5C032, 5C060, 5C090, 5C180 5C031, 5C032, 5C060, 5C090, 5C180 5C031, 5C032, 5C060, 5C090, 5C180
Logical Dev.	ALLPRO	—	V1.49—DIPs: 85C220, 85C224, 85C060, 85C090, 85C508, 5AC312, 5AC324, 5C031, 5C032, 5C060, 5C090 w/socket converters from outside source—PLCCs: 85C220, 85C224, 85C060, 85C090, 85C508, 5AC312, 5AC324, 5C060, 5C090, 5C180
Oliver Adv. Eng.	OMNI-28 OMNI-40 OMNI-64	—	DIPs: 85C220, 5C031, 5C032, 85C224, 85C060, 5AC312, 5C060 w/OM-S-20 LCC—85C220 (PLCC) w/OM-S-24 LCC—85C224 (PLCC), 85C060 (PLCC), 5C060 (PLCC) All DIPs above + DIPs: 85C090, 5AC324, 5C090, 5C121 w/OM-S-40 LCC—85C090 (PLCC), 5AC324 (PLCC), 5C090 (PLCC) All DIPs above w/OM-S-68 LCC—5C180 (PLCC)
SMS GmbH	Sprint Plus	— w/Pod	V3.2i—DIPs: 5C031, 5C032, 5C060, 5C090, 5AC312 5C180 (PLCC)
Stag	ZL-30 ZL30A ZL33 PPZ System 3000	— — w/30A640 — — —	DIPs: 5AC312, 5C031, 5C032, 5C060 DIPs: 5AC312, 5C031, 5C032, 5C060 5AC312 (PLCC), 5C060 (PLCC), 5C090 (DIP & PLCC) DIPs: 5C031, 5C032, 5C060 DIPs: 5C031, 5C032, 5C060, 5C090 DIPs: 5AC312, 5C031, 5C032, 5C060, 5C090, 5C180
System General	SGUP-85A	—	V1.8—DIPs: 85C220, 85C224, 85C508, 85C960, 5AC312, 5C031, 5C032, 5C060, 5C090 w/socket converters from outside source—PLCCs: 85C220, 85C224, 85C508, 85C960, 5AC312, 5C060, 5C090

PLD PROGRAMMING VENDORS

The following table lists programming vendors and their products that support Intel PLDs.

Vendor Information	Product	Description
Advin Systems Inc. 1050 E. Duane Ave., Suite L Sunnyvale, CA 94086 (408) 984-8600	Sailor PAL/SA Sailor PAL/SB	Univ. 28 Pins + Expansion Modules Univ. 28 Pins + Expansion Modules
BP Microsystems 10681 Haddington, #190 Houston, TX 77043 (800) 225-2102 or (713) 461-9430	PLD-1100 PLD-1128	20/24 Pins + PLCC Socket Converters 28 Pins + PLCC Socket Converter
Bytek Corp. Instrument Systems Division 508 N. W. 77th St. Boca Raton, FL 33487 (407) 994-3520	135H-U 145H-U	Universal MULTIPROGRAMMER® Logic Programmer
Data I/O Corp. 10525 Willows Road, N.E. P.O. Box 97046 Redmond, WA 98073-9746 (800) 247-5700 or (206) 881-6444	UNISITE 40/48 Model 2900 Model 29B Model 60A/H	Univ. 68 Pins—CHIPSITE for PLCC Univ. 40 Pins—DIP and PLCC Univ. 40 Pins—LogicPak for PLDs Univ. 28 Pins—H = Handler Version
Digelec Inc. 20144 Plummer St. Chatsworth, CA 91311 (800) 367-8750 or (818) 887-3755	860	PLD Programmer + PLCC Adaptor
Elan Digital Systems 2162 N. Main St. Walnut Creek, CA 94596 (415) 932-0882	1014 5-145	Univ. + PLCC Socket Converters Programmer + PLCC Socket Converters
Intel Corporation (See Sales Office and Distributor Listings at Back of Handbook.)	iUP-PC iUP-200A/201A	Intel PC-Based + GUPI Adaptors Intel Standalone + GUPI Adaptors
Kontron 630 Clyde Ave. Mountain View, CA 94039-7230 (800) 227-8834 or (415) 965-7020	EPP-80 MPP-80	Univ. + UPM Modules Univ. + UPM Modules
Logical Devices, Inc. 1201 N.W. 65th Place Ft. Lauderdale, FL 33309 (305) 974-0967	ALLPRO	Univ. 32 or 40 Pins + PLCC Socket Converters
Oliver Advanced Engineering 320 Arden St. Glendale, CA 91203 (818) 240-0080	OMNI-64 OMNI-40 OMNI-28	Univ. 64 Pins Univ. 40 Pins Univ. 28 Pins
SMS GmbH Im Morgenthal 13 D-8994 HERGATZ Germany	Sprint Plus	Univ. 28 Pin + Expansion Pods

PLD PROGRAMMING VENDORS (Continued)

The following table lists programming vendors and their products that support Intel PLDs.

Vendor Information	Product	Description
Stag Microsystems, Inc. 1600 Wyatt Drive, Suite 3 Santa Clara, CA 95054 (408) 988-1118	ZL-30 ZL-30A ZL33 PPZ	Logic 28 Pin + PLCC Adaptor Logic 28 Pin + PLCC Adaptor Gang 24 Pin Univ. 28 Pin + PLCC
System General Corp. 244 S. Hillview Dr. Milpitas, CA 95035 (408) 263-6667	SGUP-85A	Univ. 68 Pins

Univ. = Universal Programmer

PLD SOFTWARE SUPPORT

Vendor	Product	Version/Devices Supported
Data I/O	ABEL	V4.0—85C220, 85C224, 85C060, 85C090, 85C508, 5AC312, 5AC324, 5C031, 5C032, 5C060, 5C090, 5C180
	GATES	V5.0—Contact Vendor for Supported Devices
	PLDTest	V1.3—Contact Vendor for Supported Devices
Hewlitt-Packard	HP PLD Design System	V3.4— 5C031, 5C032, 5C060, 5C090, 5C180
Intel Corp.	iPLS II	V2.2 (DOS), V3.0 (SUN3/4)—85C220, 85C224, 85C060, 85C090, 85C508, 85C960, 5AC312, 5AC324, 5C031, 5C032, 5C060, 5C090, 5C180
	SCHEMA III-PLD	V1.0—85C220, 85C224, 85C060, 85C090, 85C508, 85C960, 5AC312, 5AC324, 5C031, 5C032, 5C060, 5C090, 5C180
	IPLDview-286	(Based on Workview V4.0)—85C220, 85C224, 85C060, 85C090, 85C508, 85C960, 5AC312, 5AC324, 5C031, 5C032, 5C060, 5C090, 5C180
ISDATA	Log/IC	V3.3—85C220, 85C224, 85C508, 5AC312, 5C031, 5C032, 5C060, 5C090, 5C180
Logical Devices	CUPL	V3.3—85C220, 85C224, 85C060, 85C090, 85C508, 5AC312, 5AC324, 5C031, 5C032, 5C060, 5C090, 5C180
	TESTPLA	Contact Vendor for Supported Devices
MINC Inc.	PLD Designer	V1.0—5C031, 5C032, 5C060, 5C090, 5C180
OrCAD Systems	SDT III	V3.2—85C220, 85C224, 85C060, 85C090, 85C508, 5AC312, 5AC324, 5C031, 5C032, 5C060, 5C090, 5C180
	VST	Contact Vendor for Supported Devices
Quadtree	Simulation Models	85C220, 85C224, 85C508, 85C060, 85C090, 5C032, 5C060, 5C090
Viewlogic, Inc.	Workview CAE	V4.0—85C220, 85C224, 85C060, 85C090, 85C508, 85C960, 5AC312, 5AC324, 5C031, 5C032, 5C060, 5C090, 5C180

PLD SOFTWARE VENDORS

The following table lists programming vendors and their products that support Intel PLDs.

Vendor Information	Product	Description
Data I/O Corp. 10525 Willows Road, N.E. P.O. Box 97046 Redmond, WA 98073-9746 (206) 881-6444	ABEL GATES PLDTest	Logic Compiler S/W Logic Synthesis S/W PLD Test S/W
Hewlitt-Packard Company Customer Information Center (1-800) 752-0900	HP PLD Design System	Compiler/Synthesis S/W
Intel Corporation (See Sales Office and Distributor Listings at Back of Handbook.)	iPLS II SCHEMA III-PLD iPLDview-286	Compiler S/W Schematic S/W Schematic/Simulation S/W
ISDATA GmbH Haid-und-New-Straβe7 D-7500 Karlsruhe West Germany	LOG/iC (Outside U.S. & Canada)	Compiler/Partitioner S/W
ISDATA, Inc. 800 Airport Rd. Monterey, CA 93940 (408) 373-3607	LOG/iC (U.S. & Canada)	Compiler/Partitioner S/W
Logical Devices, Inc. 1201 N.W. 65th Place Ft. Lauderdale, FL 33309 (305) 974-0967	CUPL TESTPLA	Compiler S/W PLD Test S/W
MINC, Inc. 6755 Earl Drive Colorado Springs, CO 80918 (719) 590-1155	PLDesigner	Compiler/Partitioner S/W
OrCAD Systems Corp. 3175 N.W. Aloclek Dr. Hillsboro, OR 97124-7135 (503) 690-9881	OrCAD/SDT III OrCAD/VST	Schematic S/W Simulation S/W
Quadtree 5150 E. Pacific Coast Highway Suite 320 Long Beach, CA 90804 (213) 597-8077	Simulation Models	See "Software Support" for Devices Supported; Contact Vendor for Platforms/Simulators Supported
Viewlogic, Inc. 313 Boston Post Road West Marlboro, MA 01752 (617) 480-0881	Intel PLD Kit	Intel PLD Design Library for Workview—Schematic/Simulation S/W

PAL*/GAL* TO INTEL PLD REPLACEMENT

Already in wide use throughout the electronics industry are numerous different Programmable Logic Devices. Most common PALs and GALs can be replaced or upgraded with the following Intel PLDs:

85C220

The 85C220 is a direct, drop-in replacement for most 20-pin PALs/GALs, although some PALs have an incompatible architecture. The 85C220 runs at 80 MHz with external feedback.

85C224

The 85C224 is a direct, drop-in replacement for most 24-pin PALs/GALs, although some PALS have an incompatible architecture. The 85C224 runs at 80 MHz with external feedback.

85C060

The 85C060 is NOT a drop-in replacment for any 24-pin bipolar PAL, though it can functionally replace many higher-density devices. Some modification of CLK and OE signals may be required.

85C220 As a 20-Pin PAL Replacement

100% Compatible	
10H8, -2	16R6A
12H6, -2	16R4A
14H4, -2	16L8A
16H2, -2	16RP6A
10L8, -2	16RP4A
12L6, -2	16P8A
16L8, A-2, A-4	16R8A
16R4, A-2, A-4	16RP8A
14L4, -2	16V8A
16L2, -2	18P8
16R8, A-2, A-4	18V8
16R6, A-2, A-4	
16P8, -2	
16RP8, -2	
16RP6, -2	
16RP4, -2	
16V8	

85C224 As a 24-Pin PAL Replacement

100% Compatible	
14L8	20L8A
16L6	20R8A
18L4	20R6A
20L2	20R4A
20L8	20V8
20R8	
20R6	
20R4	

85C060 As a 24-Pin PAL Replacement

Modified Replacement
20RA10
22V10
32V10
26V10
26V12

*PAL is a registered trademark of Advanced Micro Devices.
*GAL is a registered trademark of Lattice Semiconductor, Incorporated.

INTEL PLD FEATURE COMPARISON

	85C220 5C032	85C224	85C060 5C060	85C090 5C090	5C180	5AC312	5AC324
INPUTS							
Dedicated	10	14	4	12	12	10	12
Maximum	18	22	20	36	60	22	36
Input Latches/Registers						Y	Y
I/O							
Number	8	8	16	24	48	12	24
Tri-State	Y	Y	Y	Y	Y	Y	Y
Programmable Polarity	Y	Y	Y	Y	Y	Y	Y
Dual-Feedback						Y	Y
MACROCELLS	8	8	16	24	48	12	24
REGISTERS							
Number	8	8	16	24	48	12	24
Types	D	D	D/T/ RS/JK	D/T/ RS/JK	D/T/ RS/JK	D/T/ RS/JK	D/T/ RS/JK
By-Pass	Y	Y	Y	Y	Y	Y	Y
Reset to 0	Y	Y	Y	Y	Y	Y	Y
Preset to 1						Y	Y
PRODUCT TERMS							
Number	72	72	160	240	480	200	394
Allocation						Y	Y
LOCAL/GLOBAL BUSES					Y		
CLOCKS	1	1	2	2	4	2	2
Asynchronous Clocking			Y	Y	Y	Y	Y
SECURITY BIT	Y	Y	Y	Y	Y	Y	Y
TURBO BIT (LOW POWER)	Y	Y	Y	Y	Y	Y	Y

EXTENDED TEMPERATURE AND MILITARY DEVICES

Intel offers the following extended temperature and military EPLDs:

5CXXX FAMILY

TD5C032-35, 40
TP5C032-35, 40
QD5C090-50, 60
TP5C090-60
TD5C060-55
QD5C060-45, 55
LP5C060-55
QP5C060-45, 55
TP5C060-55
TN5C060-55
TD5C090-60
LD5C090-60
MD5C090-60
TN5C090-60
LD5C060-55
MD5C060-55
TN5C180-90
TA5C180-90
MQ5C180-90
MG5C180-90

5ACXXX FAMILY

TN5AC312-30, 35
MD5AC312-30, 35
TN5AC324-30, 35

85C μPLD FAMILY

TN85C220-66
TD85C220-66

For detailed information on military devices, refer to the *Military Products Handbook*, order number: 210461.

PRE-PROGRAMMED DEVICES

Intel has the capability of providing pre-programmed and specially tested PLDs. Please contact your local Intel sales representative for details.

COMPATIBLE COMPUTERS FOR iPLDS II

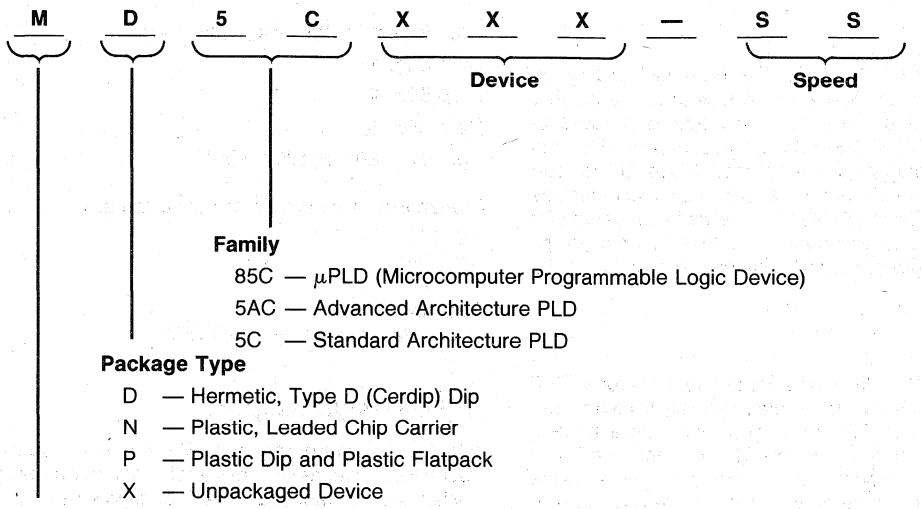
A partial list of computers that have been verified to be software compatible with the Intel Programmable Logic Development System (iPLDS II) is given below:

AT&T 6300 and 6300+
Compaq family of PCs (88, 86, 286, 386)
IBM AT
IBM XT
IBM XT-286
Intel (OEM) Systems 301/302
HP Vectra Family
Sperry IT
Tandy 3000 HD

The IBM Personal System II Models 30 (with hard disk) 50, 60, 70, and 80 can run iPLS II (Intel Programmable Logic Software)

ORDERING INFORMATION

Intel PLDs are identified as follows:



A — Indicates automotive operating temperature range (−40°C to +125°C)

J — Indicates a JAN qualified device, but is for internal identification purposes only. All JAN devices must be ordered by M38510 part number. (Example: M38510/42001 BQB), and will be marked in accordance with MIL-M-38510 specifications.

L — Indicates extended operating temperature range (−40°C to +85°C) express product with 160 + 8 hrs. dynamic burn-in.

*M — Indicates military operating temperature range (−55°C to +125°C)

Q — Indicates commercial temperature range (0°C to 70°C) express product with 160 + 8 hrs. dynamic burn-in.

T — Indicates extended temperature range (−40°C to +85°C) express product without burn-in.

— No letter indicates commercial temperature range (0°C to 70°C) without burn-in.

Examples:

QD5C060-45 Commercial with burn-in, ceramic Dip, 060 (600 gate) device, 45 nanosecond.

*On military temperature devices, B suffix indicates MIL-STD-883C level B processing.

ORDERING INFORMATION
EPLD CUSTOMER SUPPORT

Hotline

The Intel EPLD Technical Hotline is manned by application personnel every business day. The number for the United States and Canada is 1-800-323-EPLD (1-800-323-3753). Outside of the U.S. and Canada, contact your local Intel Sales Office. The Hotline is provided to assist with technical questions concerning Intel EPLDs. A recorder is connected for receiving messages during off-hours or when all applications personnel are busy handling calls.

BBS

Intel has a Bulletin Board System for registered iPLS and iPLS II customers to electronically transfer information. Any registered iPLS II user with a modem can log onto the system. The current number is (916) 985-2308. If your communication software supports file transfers, you can receive utilities, software updates, and the latest information on EPLDs via the Bulletin Board.

Data format for the BBS is as follows:

Start Bits: 1

Stop Bits: 1

Data Bits: 8

Speed: 300 or 1200 BAUD

Transmit/receive protocols supported are:

ASCII
XMODEM
KERMIT
TELINK
Cyclic Redundancy on XMODEM

EPLD Design Support

Intel has hardware designers who can help you with your EPLD designs. For more information on design assistance, contact your local Intel field sales office.

DOMESTIC SALES OFFICES

ALABAMA

Intel Corp.
5015 Bradford Dr., #2
Huntsville 35805
Tel: (205) 830-4010
FAX: (205) 837-2640

ARIZONA

†Intel Corp.
410 North 44th Street
Suite 500
Phoenix 85008
Tel: (602) 231-0386
FAX: (602) 244-0446

Intel Corp.
7225 N. Mona Lisa Rd.
Suite 215
Tucson 85741
Tel: (602) 544-0227
FAX: (602) 544-0232

CALIFORNIA

†Intel Corp.
21515 Vanowen Street
Suite 116
Canoga Park 91303
Tel: (818) 704-8500
FAX: (818) 340-1144

†Intel Corp.
300 N. Continental Blvd.
Suite 100
El Segundo 90245
Tel: (213) 640-6040
FAX: (213) 640-7133

Intel Corp.
1 Sierra Gate Plaza
Suite 280C
Roseville 95678
Tel: (916) 782-8086
FAX: (916) 782-8153

†Intel Corp.
9665 Chesapeake Dr.
Suite 325
San Diego 92123
Tel: (619) 292-8086
FAX: (619) 292-0628

†Intel Corp.*
400 N. Tustin Avenue
Suite 450
Santa Ana 92705
Tel: (714) 835-9642
TWX: 910-595-1114
FAX: (714) 541-9157

†Intel Corp.*
San Tomas 4
2700 San Tomas Expressway
2nd Floor
Santa Clara 95051
Tel: (408) 986-8086
TWX: 910-338-0255
FAX: (408) 727-2620

COLORADO

Intel Corp.
4445 Northpark Drive
Suite 100
Colorado Springs 80907
Tel: (719) 594-6622
FAX: (303) 594-0720

†Intel Corp.*
600 S. Cherry St.
Suite 700
Denver 80222
Tel: (303) 321-8086
TWX: 910-931-2289
FAX: (303) 322-8670

CONNECTICUT

†Intel Corp.
301 Lee Farm Corporate Park
83 Wooster Heights Rd.
Danbury 06810
Tel: (203) 748-3130
FAX: (203) 794-0339

FLORIDA

†Intel Corp.
800 Fairway Drive
Suite 160
Deerfield Beach 33441
Tel: (305) 421-0506
FAX: (305) 421-2444

†Intel Corp.
5850 T.G. Lee Blvd.
Suite 340
Orlando 32822
Tel: (407) 240-8000
FAX: (407) 240-8097

Intel Corp.
11300 4th Street North
Suite 170
St. Petersburg 33716
Tel: (813) 577-2413
FAX: (813) 578-1607

GEORGIA

†Intel Corp.
20 Technology Parkway
Suite 150
Norcross 30092
Tel: (404) 449-0541
FAX: (404) 605-9762

ILLINOIS

†Intel Corp.*
Woodfield Corp. Center III
300 N. Martingale Road
Suite 400
Schaumburg 60173
Tel: (708) 605-8031
FAX: (708) 706-9762

INDIANA

†Intel Corp.
8910 Purdue Road
Suite 350
Indianapolis 46268
Tel: (317) 875-0623
FAX: (317) 875-8938

IOWA

Intel Corp.
1930 St. Andrews Drive N.E.
2nd Floor
Cedar Rapids 52402
Tel: (319) 393-5510

KANSAS

†Intel Corp.
10985 Cody St.
Suite 140
Overland Park 66210
Tel: (913) 345-2727
FAX: (913) 345-2076

MARYLAND

†Intel Corp.*
10010 Junction Dr.
Suite 200
Annapolis Junction 20701
Tel: (301) 206-2860
FAX: (301) 206-3677
(301) 206-3678

MASSACHUSETTS

†Intel Corp.*
Westford Corp. Center
3 Carlisle Road
2nd Floor
Westford 01886
Tel: (508) 692-0960
TWX: 710-343-6333
FAX: (508) 692-7867

MICHIGAN

†Intel Corp.
7071 Orchard Lake Road
Suite 100
West Bloomfield 48322
Tel: (313) 851-8096
FAX: (313) 851-8770

MINNESOTA

†Intel Corp.
3500 W. 80th St.
Suite 360
Bloomington 55431
Tel: (612) 835-6722
TWX: 910-576-2867
FAX: (612) 831-6497

MISSOURI

†Intel Corp.
4203 Earth City Expressway
Suite 131
Earth City 63045
Tel: (314) 291-1990
FAX: (314) 291-4341

NEW JERSEY

†Intel Corp.*
Lincroft Office Center
125 Half Mile Road
Red Bank 07701
Tel: (908) 747-2233
FAX: (908) 747-0983

Intel Corp.
280 Corporate Center
75 Livingston Avenue
First Floor
Roseland 07068
Tel: (201) 740-0111
FAX: (201) 740-0626

NEW YORK

Intel Corp.*
850 Crosskeys Office Park
Fairport 14450
Tel: (716) 425-2750
TWX: 510-253-7391
FAX: (716) 223-2561

†Intel Corp.*
2950 Express Dr., South
Suite 130
Islandia 11722
Tel: (516) 231-3300
TWX: 510-227-6236
FAX: (516) 348-7939

†Intel Corp.
300 Westage Business Center
Suite 230
Fishkill 12524
Tel: (914) 897-3860
FAX: (914) 897-3125

Intel Corp.
Seventeen State Street
14th Floor
New York 10004
Tel: (212) 248-8086
FAX: (212) 248-0888

NORTH CAROLINA

†Intel Corp.
5800 Executive Center Dr.
Suite 105
Charlotte 28212
Tel: (704) 568-8966
FAX: (704) 535-2236

†Intel Corp.
5540 Centerview Dr.
Suite 215
Raleigh 27606
Tel: (919) 851-9537
FAX: (919) 851-8974

OHIO

†Intel Corp.*
3401 Park Center Drive
Suite 220
Dayton 45414
Tel: (513) 890-5350
TWX: 810-450-2528
FAX: (513) 890-8658

†Intel Corp.*
25700 Science Park Dr.
Suite 100
Beachwood 44122
Tel: (216) 464-2736
TWX: 810-427-9298
FAX: (804) 282-0673

OKLAHOMA

Intel Corp.
6801 N. Broadway
Suite 115
Oklahoma City 73162
Tel: (405) 848-8086
FAX: (405) 840-9819

OREGON

†Intel Corp.
15254 N.W. Greenbrier Pkwy.
Building B
Beaverton 97006
Tel: (503) 645-8051
TWX: 910-467-8741
FAX: (503) 645-8181

PENNSYLVANIA

†Intel Corp.*
925 Harvest Drive
Suite 200
Blue Bell 19422
Tel: (215) 641-1000
FAX: (215) 641-0785

†Intel Corp.*
400 Penn Center Blvd.
Suite 610
Pittsburgh 15235
Tel: (412) 823-4970
FAX: (412) 829-7578

PUERTO RICO

†Intel Corp.
South Industrial Park
P.O. Box 910
Las Piedras 00671
Tel: (809) 733-8616

TEXAS

Intel Corp.
8911 N. Capital of Texas Hwy.
Suite 4230
Austin 78759
Tel: (512) 794-8086
FAX: (512) 338-9335

†Intel Corp.*
12000 Ford Road
Suite 400
Dallas 75234
Tel: (214) 241-8087
FAX: (214) 484-1180

†Intel Corp.*
7322 S.W. Freeway
Suite 1490
Houston 77074
Tel: (713) 988-8086
TWX: 910-881-2490
FAX: (713) 988-3660

UTAH

†Intel Corp.
428 East 6400 South
Suite 104
Murray 84107
Tel: (801) 263-8051
FAX: (801) 268-1457

VIRGINIA

†Intel Corp.
9030 Stony Point Pkwy.
Suite 360
Richmond 23235
Tel: (804) 330-9393
FAX: (804) 330-3019

WASHINGTON

†Intel Corp.
155 108th Avenue N.E.
Suite 386
Bellevue 98004
Tel: (206) 453-8086
TWX: 910-443-3002
FAX: (206) 451-9556

Intel Corp.
408 N. Mullan Road
Suite 102
Spokane 99206
Tel: (509) 928-8086
FAX: (509) 928-9467

WISCONSIN

Intel Corp.
330 S. Executive Dr.
Suite 102
Brookfield 53005
Tel: (414) 784-8087
FAX: (414) 796-2115

CANADA

BRITISH COLUMBIA

Intel Semiconductor of
Canada, Ltd.
4585 Canada Way
Suite 202
Burnaby V5G 4L6
Tel: (604) 298-0387
FAX: (604) 298-8234

ONTARIO

†Intel Semiconductor of
Canada, Ltd.
2650 Queensview Drive
Suite 250
Ottawa K2B 8H6
Tel: (613) 829-9714
FAX: (613) 820-5936

†Intel Semiconductor of
Canada, Ltd.
190 Attwell Drive
Suite 500
Rexdale M9W 6H8
Tel: (416) 675-2105
FAX: (416) 675-2438

QUEBEC

†Intel Semiconductor of
Canada, Ltd.
1 Rue Holiday
Suite 115
Tour East
Pt. Claire H9R 5N3
Tel: (514) 694-9130
FAX: 514-694-0064

†Sales and Service Office
*Field Application Location

DOMESTIC DISTRIBUTORS

ALABAMA

Arrow Electronics, Inc.
1015 Henderson Road
Huntsville 35805
Tel: (205) 837-6955
FAX: 205-751-1581

Hamilton/Avnet Computer
4930 I Corporate Drive
Huntsville 35805

Hamilton/Avnet Electronics
4940 Research Drive
Huntsville 35805
Tel: (205) 837-7210
FAX: 205-721-0356

MTI Systems Sales
4950 Corporate Drive
Suite 120
Huntsville 35806
Tel: (205) 830-9526
FAX: (205) 830-9557

Pioneer/Technologies Group, Inc.
4825 University Square
Huntsville 35805
Tel: (205) 837-9300
FAX: 205-837-9358

ALASKA

Hamilton/Avnet Computer
1400 W. Benson Blvd., Suite 400
Anchorage 99503

ARIZONA

†Arrow Electronics, Inc.
4134 E. Wood Street
Phoenix 85040
Tel: (602) 437-0750
TWX: 910-951-1550

Hamilton/Avnet Computer
30 South McKemy Avenue
Chandler 85226

Hamilton/Avnet Computer
90 South McKemy Road
Chandler 85226

†Hamilton/Avnet Electronics
505 S. Madison Drive
Tempe 85281
Tel: (602) 231-5140
TWX: 910-950-0077

Hamilton/Avnet Electronics
30 South McKemy
Chandler 85226
Tel: (602) 961-6669
FAX: 602-961-4073

Wyle Distribution Group
4141 E. Raymond
Phoenix 85040
Tel: (602) 249-2232
TWX: 910-371-2871

CALIFORNIA

Arrow Commercial System Group
1502 Crocker Avenue
Hayward 94544
Tel: (415) 489-5371
FAX: (415) 489-9393

Arrow Commercial System Group
14242 Chambers Road
Tustin 92680
Tel: (714) 544-0200
FAX: (714) 731-8438

†Arrow Electronics, Inc.
19748 Dearborn Street
Chatsworth 91311
Tel: (213) 701-7500
TWX: 910-493-2086

†Arrow Electronics, Inc.
9511 Ridgehaven Court
San Diego 92123
Tel: (619) 565-4800
FAX: 619-279-8062

†Arrow Electronics, Inc.
521 Weddell Drive
Sunnyvale 94086
Tel: (408) 745-6600
TWX: 910-339-9371

†Arrow Electronics, Inc.
2961 Dow Avenue
Tustin 92680
Tel: (714) 838-5422
TWX: 910-595-2860

Hamilton/Avnet Computer
3170 Pullman Street
Costa Mesa 92626

Hamilton/Avnet Computer
1361B West 190th Street
Gardena 90248

Hamilton/Avnet Computer
4103 Northgate Blvd.
Sacramento 95834

Hamilton/Avnet Computer
4545 Viewridge Avenue
San Diego 92123

Hamilton/Avnet Computer
1175 Bordeaux Drive
Sunnyvale 94089

Hamilton/Avnet Electronics
21150 Califa Street
Woodland Hills 91367

†Hamilton/Avnet Electronics
3170 Pullman Street
Costa Mesa 92626
Tel: (714) 641-4150
TWX: 910-595-2638

†Hamilton/Avnet Electronics
1175 Bordeaux Drive
Sunnyvale 94086
Tel: (408) 743-3300
TWX: 910-339-9332

†Hamilton/Avnet Electronics
4545 Ridgeview Avenue
San Diego 92123
Tel: (619) 571-7500
TWX: 910-595-2638

†Hamilton/Avnet Electronics
21150 Califa St.
Woodland Hills 91376
Tel: (818) 594-0404
FAX: 818-594-8233

†Hamilton/Avnet Electronics
10950 W. Washington Blvd.
Culver City 20230
Tel: (213) 558-2458
TWX: 910-340-6364

†Hamilton/Avnet Electronics
1361B West 190th Street
Gardena 90248
Tel: (213) 217-6700
TWX: 910-340-6364

†Hamilton/Avnet Electronics
4103 Northgate Blvd.
Sacramento 95834
Tel: (916) 920-3150

Pioneer/Technologies Group, Inc.
134 Rio Robles
San Jose 95134
Tel: (408) 954-9100
FAX: 408-954-9113

Wyle Distribution Group
124 Maryland Street
El Segundo 90254
Tel: (213) 322-8100

Wyle Distribution Group
7431 Chapman Ave.
Garden Grove 92641
Tel: (714) 891-1717
FAX: 714-891-1621

†Wyle Distribution Group
2951 Sunrise Blvd., Suite 175
Rancho Cordova 95742
Tel: (916) 638-5282

†Wyle Distribution Group
9525 Chesapeake Drive
San Diego 92123
Tel: (619) 565-9171
TWX: 910-335-1590

†Wyle Distribution Group
3000 Bowers Avenue
Santa Clara 95051
Tel: (408) 727-2500
TWX: 408-988-2747

†Wyle Distribution Group
17872 Cowan Avenue
Irvine 92714
Tel: (714) 863-9953
TWX: 910-371-7127

†Wyle Distribution Group
26677 W. Agoura Rd.
Calabasas 91302
Tel: (818) 880-9000
TWX: 372-0232

COLORADO

Arrow Electronics, Inc.
7060 South Tucson Way
Englewood 80112
Tel: (303) 790-4444

Hamilton/Avnet Computer
9605 Maroon Circle, Ste. 200
Engelwood 80112

†Hamilton/Avnet Electronics
9605 Maroon Circle
Suite 200
Englewood 80112
Tel: (303) 799-0663
TWX: 910-935-0787

†Wyle Distribution Group
451 E. 124th Avenue
Thornton 80241
Tel: (303) 457-9953
TWX: 910-936-0770

CONNECTICUT

†Arrow Electronics, Inc.
12 Beaumont Road
Wallingford 06492
Tel: (203) 265-7741
TWX: 710-476-0162

Hamilton/Avnet Computer
Commerce Industrial Park
Commerce Drive
Danbury 06810

†Hamilton/Avnet Electronics
Commerce Industrial Park
Commerce Drive
Danbury 06810
Tel: (203) 797-2800
TWX: 710-456-9974

†Pioneer/Standard Electronics
112 Main Street
Norwalk 06851
Tel: (203) 853-1515
FAX: 203-838-9901

FLORIDA

†Arrow Electronics, Inc.
400 Fairway Drive
Suite 102
Deerfield Beach 33441
Tel: (305) 429-8200
FAX: 305-428-3991

†Arrow Electronics, Inc.
37 Skyline Drive
Suite 3101
Lake Marv 32746
Tel: (407) 323-0252
FAX: 407-323-3189

Hamilton/Avnet Computer
6801 N.W. 15th Way
Ft. Lauderdale 33309

Hamilton/Avnet Computer
3247 Spring Forest Road
St. Petersburg 33702

†Hamilton/Avnet Electronics
6801 N.W. 15th Way
Ft. Lauderdale 33309
Tel: (305) 971-2900
FAX: 305-971-5420

†Hamilton/Avnet Electronics
3197 Tech Drive North
St. Petersburg 33702
Tel: (813) 573-3930
FAX: 813-572-4329

†Hamilton/Avnet Electronics
6947 University Boulevard
Winter Park 32792
Tel: (407) 628-3888
FAX: 407-678-1878

†Pioneer/Technologies Group, Inc.
337 Northlake Blvd., Suite 1000
Alta Monte Springs 32701
Tel: (407) 834-9090
FAX: 407-834-0865

Pioneer/Technologies Group, Inc.
674 S. Military Trail
Deerfield Beach 33442
Tel: (305) 428-8877
FAX: 305-481-2950

GEORGIA

Arrow Commercial System Group
3400 C. Corporate Way
Deluth 30139
Tel: (404) 623-8825
FAX: (404) 623-8802

†Arrow Electronics, Inc.
4250 E. Rivergreen Parkway
Deluth 30136
Tel: (404) 497-1300
TWX: 810-766-0439

Hamilton/Avnet Computer
5825 D. Peachtree Corners E.
Norcross 30092

†Hamilton/Avnet Electronics
5825 D Peachtree Corners
Norcross 30092
Tel: (404) 447-7500
TWX: 810-766-0432

Pioneer/Technologies Group, Inc.
3100 F Northwoods Place
Norcross 30071
Tel: (404) 448-1711
FAX: 404-446-8270

ILLINOIS

†Arrow Electronics, Inc.
1140 W. Thorndale
Itasca 60143
Tel: (708) 250-0500
TWX: 708-250-0916

Hamilton/Avnet Computer
1130 Thorndale Avenue
Bensenville 60106

†Hamilton/Avnet Electronics
1130 Thorndale Avenue
Bensenville 60106
Tel: (708) 860-7780
TWX: 708-860-8530

MTI Systems Sales
1100 W. Thorndale
Itasca 60143
Tel: (708) 773-2300

†Pioneer/Standard Electronics
2171 Executive Dr., Suite 200
Addison 60101
Tel: (708) 495-9680
FAX: 708-495-9831

INDIANA

†Arrow Electronics, Inc.
7108 Lakeview Parkway West Drive
Indianapolis 46268
Tel: (317) 299-2071
FAX: 317-299-0255

Hamilton/Avnet Computer
485 Gradle Drive
Carmel 46032

Hamilton/Avnet Electronics
485 Gradle Drive
Carmel 46032
Tel: (317) 844-9333
FAX: 317-844-5921

†Pioneer/Standard Electronics
9350 Priority Way
West Drive
Indianapolis 46250
Tel: (317) 573-0880
FAX: 317-573-0979

†Certified Technical Distributor

DOMESTIC DISTRIBUTORS (Contd.)

IOWA

Hamilton/Avnet Computer
915 33rd Avenue SW
Cedar Rapids 52404

Hamilton/Avnet Electronics
915 33rd Avenue, S.W.
Cedar Rapids 52404
Tel: (319) 362-4757

KANSAS

Arrow Electronics, Inc.
8208 Melrose Dr., Suite 210
Lenexa 66214
Tel: (913) 541-9542
FAX: 913-541-0328

Hamilton/Avnet Computer
15313 W. 95th Street
Lenexa 61219

†Hamilton/Avnet Electronics
15313 W. 95th
Overland Park 66215
Tel: (913) 888-8900
FAX: 913-541-7951

KENTUCKY

Hamilton/Avnet Electronics
805 A. Newtown Circle
Lexington 40511
Tel: (606) 259-1475

MARYLAND

†Arrow Electronics, Inc.
8300 Guilford Drive
Suite H, River Center
Columbia 21046
Tel: (301) 995-6002
FAX: 301-381-3854

Hamilton/Avnet Computer
6822 Oak Hall Lane
Columbia 21045

†Hamilton/Avnet Electronics
6822 Oak Hall Lane
Columbia 21045
Tel: (301) 995-3500
FAX: 301-995-3593

†Mesa Technology Corp.
9720 Patuxent Woods Dr.
Columbia 21046
Tel: (301) 290-8150
FAX: 301-290-6474

†Pioneer/Technologies Group, Inc.
9100 Gaither Road
Gaithersburg 20877
Tel: (301) 921-0660
FAX: 301-921-4255

MASSACHUSETTS

Arrow Electronics, Inc.
25 Upton Dr.
Wilmington 01887
Tel: (508) 658-0900
TWX: 710-393-6770

Hamilton/Avnet Computer
10 D Centennial Drive
Peabody 01960

†Hamilton/Avnet Electronics
10D Centennial Drive
Peabody 01960
Tel: (508) 532-9838
FAX: 508-596-7802

†Pioneer/Standard Electronics
44 Hartwell Avenue
Lexington 02173
Tel: (617) 861-9200
FAX: 617-863-1547

Wyle Distribution Group
15 Third Avenue
Burlington 01803
Tel: (617) 272-7300
FAX: 617-272-6809

MICHIGAN

†Arrow Electronics, Inc.
19880 Haggerty Road
Livonia 48152
Tel: (313) 665-4100
TWX: 810-223-6020

Hamilton/Avnet Computer
2215 S.E. A-5
Grand Rapids 49508

Hamilton/Avnet Computer
41650 Garden Rd., Ste. 100
Novi 48050

Hamilton/Avnet Electronics
2215 29th Street S.E.
Space A5
Grand Rapids 49508
Tel: (616) 243-8805
FAX: 616-698-1831

Hamilton/Avnet Electronics
41650 Garden Brook
Novi 48050
Tel: (313) 347-4271
FAX: 313-347-4021

†Pioneer/Standard Electronics
4505 Broadmoor S.E.
Grand Rapids 49508
Tel: (616) 698-1800
FAX: 616-698-1831

†Pioneer/Standard Electronics
13485 Stamford
Livonia 48150
Tel: (313) 525-1800
FAX: 313-427-3720

MINNESOTA

†Arrow Electronics, Inc.
5230 W. 73rd Street
Edina 55435
Tel: (612) 830-1800
TWX: 910-576-3125

Hamilton/Avnet Computer
12400 Whitewater Drive
Minnetonka 55343

†Hamilton/Avnet Electronics
12400 Whitewater Drive
Minnetonka 55434
Tel: (612) 932-0600
TWX: 910-576-2720

†Pioneer/Standard Electronics
7625 Golden Triange Dr.
Suite G
Eden Prairie 55343
Tel: (612) 944-3355
FAX: 612-944-3794

MISSOURI

†Arrow Electronics, Inc.
2380 Schuetz
St. Louis 63141
Tel: (314) 567-6888
FAX: 314-567-1164

Hamilton/Avnet Computer
739 Goddard Avenue
Chesterfield 63005

†Hamilton/Avnet Electronics
741 Goddard
Chesterfield 63005
Tel: (314) 537-1600
FAX: 314-537-4248

NEW HAMPSHIRE

Hamilton/Avnet Computer
2 Executive Park Drive
Bedford 03102

Hamilton/Avnet Computer
444 East Industrial Park Dr.
Manchester 03103

NEW JERSEY

†Arrow Electronics, Inc.
4 East Stow Road
Unit 11
Marlton 08053
Tel: (609) 596-8000
FAX: 609-596-9632

†Arrow Electronics
6 Century Drive
Parsipanny 07054
Tel: (201) 538-0900
FAX: 201-538-0900

Hamilton/Avnet Computer
1 Keystone Ave., Bldg. 36
Cherry Hill 08003

Hamilton/Avnet Computer
10 Industrial Road
Fairfield 07006

†Hamilton/Avnet Electronics
1 Keystone Ave., Bldg. 36
Cherry Hill 08003
Tel: (609) 424-0110
FAX: 609-751-2552

†Hamilton/Avnet Electronics
10 Industrial
Fairfield 07006
Tel: (201) 575-3390
FAX: 201-575-5839

†MTI Systems Sales
9 Law Drive
Fairfield 07006
Tel: (201) 227-5552
FAX: 201-575-6336

†Pioneer/Standard Electronics
14-A Madison Rd.
Fairfield 07006
Tel: (201) 575-3510
FAX: 201-575-3454

NEW MEXICO

Alliance Electronics Inc.
10510 Research Avenue
Albuquerque 87123
Tel: (505) 292-3360
FAX: 505-292-6537

Hamilton/Avnet Computer
5659 Jefferson, N.E. Suites A & B
Albuquerque 87109

†Hamilton/Avnet Electronics
5659A Jefferson N.E.
Albuquerque 87109
Tel: (505) 765-1500
FAX: 505-243-1395

NEW YORK

†Arrow Electronics, Inc.
3375 Brighton Henrietta Townline Rd.
Rochester 14623
Tel: (716) 427-0300
TWX: 510-253-4766

Arrow Electronics, Inc.
20 Oser Avenue
Hauppauge 11788
Tel: (516) 231-1000
TWX: 510-227-6623

Hamilton/Avnet Computer
933 Motor Parkway
Haupauge 11788

Hamilton/Avnet Computer
2060 Townline
Rochester 14623

†Hamilton/Avnet Electronics
933 Motor Parkway
Hauppauge 11788
Tel: (516) 231-9800
TWX: 510-224-6166

†Hamilton/Avnet Electronics
2060 Townline Rd.
Rochester 14623
Tel: (716) 272-2744
TWX: 510-253-5470

Hamilton/Avnet Electronics
103 Twin Oaks Drive
Syracuse 13206
Tel: (315) 437-0288
TWX: 710-541-1560

†MTI Systems Sales
38 Harbor Park Drive
Port Washington 11050
Tel: (516) 621-6200
FAX: 510-223-0846

Pioneer/Standard Electronics
68 Corporate Drive
Binghamton 13904
Tel: (607) 722-9300
FAX: 607-722-9562

Pioneer/Standard Electronics
40 Oser Avenue
Hauppauge 11787
Tel: (516) 231-9200
FAX: 510-227-9869

†Pioneer/Standard Electronics
60 Crossway Park West
Woodbury, Long Island 11797
Tel: (516) 921-8700
FAX: 516-921-2143

†Pioneer/Standard Electronics
840 Fairport Park
Fairport 14450
Tel: (716) 381-7070
FAX: 716-381-5955

NORTH CAROLINA

†Arrow Electronics, Inc.
5240 Greensdairy Road
Raleigh 27604
Tel: (919) 876-3132
TWX: 510-928-1856

Hamilton/Avnet Computer
3510 Spring Forest Road
Raleigh 27604

†Hamilton/Avnet Electronics
3510 Spring Forest Drive
Raleigh 27604
Tel: (919) 878-0819
TWX: 510-928-1836

Pioneer/Technologies Group, Inc.
9401 L-Southern Pine Blvd.
Charlotte 28210
Tel: (919) 527-8188
FAX: 704-522-8564

Pioneer Technologies Group, Inc.
2810 Meridian Parkway
Suite 148
Durham 27713
Tel: (919) 544-5400
FAX: 919-544-5885

OHIO

Arrow Commercial System Group
284 Cramer Creek Court
Dublin 43017
Tel: (614) 889-9347
FAX: (614) 889-9680

†Arrow Electronics, Inc.
6238 Cochran Road
Solon 44139
Tel: (216) 248-3990
TWX: 810-427-9409

Hamilton/Avnet Computer
7764 Washington Village Dr.
Dayton 45459

Hamilton/Avnet Computer
30325 Bainbridge Rd., Bldg. A
Solon 44139

†Hamilton/Avnet Electronics
7760 Washington Village Dr.
Dayton 45459
Tel: (513) 439-6733
FAX: 513-439-6711

†Hamilton/Avnet Electronics
30325 Bainbridge
Solon 44139
Tel: (216) 349-5100
TWX: 810-427-9452

Hamilton/Avnet Computer
777 Brooksedge Blvd.
Westerville 43081
Tel: (614) 882-7004
FAX: 614-882-8650

Hamilton/Avnet Electronics
777 Brooksedge Blvd.
Westerville 43081
Tel: (614) 882-7004

MTI Systems Sales
23400 Commerce Park Road
Beachwood 44122
Tel: (216) 464-6688

†Pioneer/Standard Electronics
4433 Interpoint Boulevard
Dayton 45424
Tel: (513) 236-9900
FAX: 513-236-8133

†Pioneer/Standard Electronics
4800 E. 131st Street
Cleveland 44105
Tel: (216) 587-3600
FAX: 216-663-1004

†Certified Technical Distributor

DOMESTIC DISTRIBUTORS (Contd.)

OKLAHOMA

Arrow Electronics, Inc.
4719 South Memorial Dr.
Tulsa 74145

†Hamilton/Avnet Electronics
12121 E. 51st St., Suite 102A
Tulsa 74146
Tel: (918) 252-7297

OREGON

†Almac Electronics Corp.
1885 N.W. 169th Place
Beaverton 97005
Tel: (503) 629-8090
FAX: 503-645-0611

Hamilton/Avnet Computer
9409 Southwest Nimbus Ave.
Beaverton 97005

†Hamilton/Avnet Electronics
9409 S.W. Nimbus Ave.
Beaverton 97005
Tel: (503) 627-0201
FAX: 503-641-4012

Wyle
9640 Sunshine Court
Bldg. G, Suite 200
Beaverton 97005
Tel: (503) 643-7900
FAX: 503-646-5466

PENNSYLVANIA

Arrow Electronics, Inc.
650 Seco Road
Monroeville 15146
Tel: (412) 856-7000

Hamilton/Avnet Computer
2800 Liberty Ave., Bldg. E
Pittsburgh 15222

Hamilton/Avnet Electronics
2800 Liberty Ave.
Pittsburgh 15238
Tel: (412) 281-4150

Pioneer/Standard Electronics
259 Kappa Drive
Pittsburgh 15238
Tel: (412) 782-2300
FAX: 412-963-8255

†Pioneer/Technologies Group, Inc.
Delaware Valley
261 Gibralter Road
Horsham 19044
Tel: (215) 674-4000
FAX: 215-674-3107

TENNESSEE

Arrow Commercial System Group
3635 Knight Road
Suite 7
Memphis 38118
Tel: (901) 367-0540
FAX: (901) 367-2081

TEXAS

Arrow Electronics, Inc.
3220 Commander Drive
Carrollton 75006
Tel: (214) 380-6464
FAX: (214) 248-7208

Hamilton/Avnet Computer
1807A West Braker Lane
Austin 78758

Hamilton/Avnet Computer
Forum 2
4004 Beltline, Suite 200
Dallas 75244

Hamilton/Avnet Computer
4850 Wright Rd., Suite 190
Stafford 77477

†Hamilton/Avnet Electronics
1807 W. Braker Lane
Austin 78758
Tel: (512) 837-8911
TWX: 910-874-1319

†Hamilton/Avnet Electronics
4004 Beltline, Suite 200
Dallas 75234
Tel: (214) 308-8111
TWX: 910-860-5929

†Hamilton/Avnet Electronics
4850 Wright Rd., Suite 190
Stafford 77477
Tel: (713) 240-7733
TWX: 910-881-5523

†Pioneer/Standard Electronics
1826-D Kramer
Austin 78758
Tel: (512) 835-4000
FAX: 512-835-9829

†Pioneer/Standard Electronics
13710 Omega Road
Dallas 75244
Tel: (214) 386-7300
FAX: 214-490-6419

†Pioneer/Standard Electronics
10530 Rockley Road
Houston 77099
Tel: (713) 495-4700
FAX: 713-495-5642

†Wyle Distribution Group
1810 Greenville Avenue
Richardson 75081
Tel: (214) 235-9953
FAX: 214-644-5064

UTAH

Hamilton/Avnet Computer
1585 West 2100 South
Salt Lake City 84119

†Hamilton/Avnet Electronics
1585 West 2100 South
Salt Lake City 84119
Tel: (801) 972-2800
TWX: 910-925-4018

†Wyle Distribution Group
1325 West 2200 South
Suite E
West Valley 84119
Tel: (801) 974-9953

WASHINGTON

†Almac Electronics Corp.
14360 S.E. Eastgate Way
Bellevue 98007
Tel: (206) 643-9992
FAX: 206-643-9709

Hamilton/Avnet Computer
17761 Northeast 78th Place
Redmond 98052

†Hamilton/Avnet Electronics
17761 N.E. 78th Place
Redmond 98052
Tel: (206) 881-6697
FAX: 206-867-0159

Wyle Distribution Group
15385 N.E. 90th Street
Redmond 98052
Tel: (206) 881-1150
FAX: 206-881-1567

WISCONSIN

Arrow Electronics, Inc.
200 N. Patrick Blvd., Ste. 100
Brookfield 53005
Tel: (414) 792-0150
FAX: 414-792-0156

Hamilton/Avnet Computer
20875 Crossroads Circle
Suite 400
Waukesha 53186

†Hamilton/Avnet Electronics
20875 Crossroads Circle
Suite 400
Waukesha 53186
Tel: (414) 784-4510
FAX: 414-784-9509

CANADA

ALBERTA

Hamilton/Avnet Computer
2816 21st Street Northeast
Calgary T2E 6Z2

Hamilton/Avnet Electronics
2816 21st Street N.E. #3
Calgary T2E 6Z3
Tel: (403) 230-3586
FAX: 403-250-1591

Zentronics
6815 #8 Street N.E.
Suite 100
Calgary T2E 7H
Tel: (403) 295-8818
FAX: 403-295-8714

BRITISH COLUMBIA

†Hamilton/Avnet Electronics
8610 Commerce Ct.
Burnaby V5A 4N6
Tel: (604) 420-4101
FAX: 604-437-4712

Zentronics
108-11400 Bridgeport Road
Richmond V6X 1T2
Tel: (604) 273-5575
FAX: 604-273-2413

ONTARIO

Arrow Electronics, Inc.
36 Antares Dr., Unit 100
Nepean K2E 7W5
Tel: (613) 226-6903
FAX: 613-723-2018

†Arrow Electronics, Inc.
1093 Meyerside, Unit 2
Mississauga L5T 1M4
Tel: (416) 673-7769
FAX: 416-672-0849

Hamilton/Avnet Computer
Canada System Engineering Group
3688 Nashua Drive
Units 7 & 8
Mississauga L4V 1M5

Hamilton/Avnet Computer
3688 Nashua Drive
Units 9 & 10
Mississauga L4V 1M5

Hamilton/Avnet Computer
6845 Rexwood Road
Units 7, 8, & 9
Mississauga L4V 1R2

Hamilton/Avnet Computer
190 Colonade Road
Nepean K2E 7J5

†Hamilton/Avnet Electronics
6845 Rexwood Road
Units 3-4-5
Mississauga L4T 1R2
Tel: (416) 677-7432
FAX: 416-677-0940

†Hamilton/Avnet Electronics
190 Colonnade Road South
Nepean K2E 7L5
Tel: (613) 226-1700
FAX: 613-226-1184

†Zentronics
1355 Meyerside Drive
Mississauga L5T 1C9
Tel: (416) 564-9600
FAX: 416-564-8320

†Zentronics
155 Colonnade Road
Unit 17
Nepean K2E 7K1
Tel: (613) 226-8840
FAX: 613-226-6352

QUEBEC

Arrow Electronics Inc.
1100 St. Regis
Dorval H9P 2T5
Tel: (514) 421-7411
FAX: 514-421-7430

Arrow Electronics, Inc.
500 Boul. St-Jean-Baptiste
Suite 280
Quebec G2E 5R9
Tel: (418) 871-7500
FAX: 418-871-6816

Hamilton/Avnet Computer
2795 Rue Halpern
St. Laurent H4S 1P8

†Hamilton/Avnet Electronics
2795 Halpern
St. Laurent H2E 7K1
Tel: (514) 335-1000
FAX: 514-335-2481

†Zentronics
520 McCaffrey
St. Laurent H4T 1N3
Tel: (514) 737-9700
FAX: 514-737-5212

†Certified Technical Distributor

EUROPEAN SALES OFFICES

FINLAND

Intel Finland OY
Ruosilantie 2
00390 Helsinki
Tel: (358) 0 544 644
TLX: 123332

FRANCE

Intel Corporation S.A.R.L.
1, Rue Edison-BP 303
78054 St. Quentin-en-Yvelines Cedex
Tel: (33) (1) 30 57 70 00
TLX: 699016

ISRAEL

Intel Semiconductor Ltd.
Atidim Industrial Park-Neve Sharet
P.O. Box 43202
Tel-Aviv 61430
Tel: (972) 03-498080
TLX: 371215

ITALY

Intel Corporation Italia S.p.A.
Milanofiori Palazzo E
20094 Assago
Milano
Tel: (39) (02) 89200950
TLX: 341286

NETHERLANDS

Intel Semiconductor B.V.
Postbus 84130
3099 CC Rotterdam
Tel: (31) 10.407.11.11
TLX: 22283

SPAIN

Intel Iberia S.A.
Zurbaran, 28
28010 Madrid
Tel: (34) (1) 308.25.52
TLX: 46880

SWEDEN

Intel Sweden A.B.
Dalvagen 24
171 36 Solna
Tel: (46) 8 734 01 00
TLX: 12261

SWITZERLAND

Intel Semiconductor A.G.
Zuerichstrasse
8185 Winkel-Rueti bei Zuerich
Tel: (41) 01/860 62 62
TLX: 825977

UNITED KINGDOM

Intel Corporation (U.K.) Ltd.
Pipers Way
Swindon, Wiltshire SN3 1RJ
Tel: (44) (0793) 696000
TLX: 444447/8

WEST GERMANY

Intel GmbH
Dornacher Strasse 1
8016 Feldkirchen bei Muenchen
Tel: (49) 089/90992-0
FAX: (49) 089/904/3948

Intel GmbH
Abraham Lincoln Strasse 16-18
6200 Wiesbaden
Tel: (49) 06121/7605-0
TLX: 4-186183

Intel GmbH
Zettachring 10A
7000 Stuttgart 80
Tel: (49) 0711/7287-280
TLX: 7-254826

EUROPEAN DISTRIBUTORS/REPRESENTATIVES

AUSTRIA

Bacher Electronics G.m.b.H.
Rotenmuehlgasse 26
1120 Wien
Tel: (43) (0222) 83 56 46
TLX: 31532

BELGIUM

Inelco Belgium S.A.
Av. des Croix de Guerre 94
1120 Bruxelles
Oorlogskruisenlaan, 94
1120 Brussel
Tel: (32) (02) 216 01 60
TLX: 64475 or 22090

DENMARK

ITT-Multikomponent
Naverland 29
2600 Glostrup
Tel: (45) (0) 2 45 66 45
TLX: 33 355

FINLAND

OY Fintronic AB
Melkonkatu 24A
00210 Helsinki
Tel: (358) (0) 6926022
TLX: 124224

FRANCE

Almex
Zone industrielle d'Antony
48, rue de l'Aubepine
BP 102
92164 Antony cedex
Tel: (33) (1) 46 66 21 12
TLX: 250067

Jermyn
60, rue des Gemeaux
Silic 580
94653 Rungis Cedex
Tel: (33) (1) 49 78 49 78
TLX: 261585

Metrologie
Tour d'Asnieres
4, av. Laurent-Cely
92606 Asnieres Cedex
Tel: (33) (1) 47 90 62 40
TLX: 611448

Tekelec-Airtronic
Cite des Bruyeres
Rue Carle Vernet - BP 2
92310 Sevres
Tel: (33) (1) 45 34 75 35
TLX: 204552

IRELAND

Micro Marketing Ltd.
Glenageary Office Park
Glenageary
Co. Dublin
Tel: (21) (353) (01) 856288
FAX: (21) (353) (01) 857364
TLX: 31584

ISRAEL

Eastronics Ltd.
11 Rozanis Street
P.O.B. 39300
Tel-Aviv 61392
Tel: (972) 03-475151
TLX: 33638

ITALY

Intesi
Divisione ITT Industries GmbH
Viale Milanofiori
Palazzo E/5
20090 Assago (MI)
Tel: (39) 02/824701
TLX: 311351

Lasi Elettronica S.p.A.
V. le Fulvio Testi, 126
20092 Cinisello Balsamo (MI)
Tel: (39) 02/2440012
TLX: 352040

Telcom S.r.l.
Via M. Civitali 75
20148 Milano
Tel: (39) 02/4049046
TLX: 335654

ITT Multicomponents
Viale Milanofiori E/5
20090 Assago (MI)
Tel: (39) 02/824701
TLX: 311351

Silverstar
Via Dei Gracchi 20
20146 Milano
Tel: (39) 02/49961
TLX: 332189

NETHERLANDS

Koning en Hartman
Elektrotechniek B.V.
Energieweg 1
2627 AP Delft
Tel: (31) (1) 15/609906
TLX: 38250

NORWAY

Nordisk Elektronikk (Norge) A/S
Postboks 123
Smedsvingen 4
1364 Hvalstad
Tel: (47) (02) 84 62 10
TLX: 77546

PORTUGAL

ATD Portugal LDA
Rua Dos Lusiados, 5 Sala B
1300 Lisboa
Tel: (35) (1) 64 80 91
TLX: 61562

Ditram
Avenida Miguel Bombarda, 133
1000 Lisboa
Tel: (35) (1) 54 53 13
TLX: 14182

SPAIN

ATD Electronica, S.A.
Plaza Ciudad de Viena, 6
28040 Madrid
Tel: (34) (1) 234 40 00
TLX: 42477

Metrologia Iberica, S.A.
Ctra. de Fuencarral, n.80
28100 Alcobendas (Madrid)
Tel: (34) (1) 653 86 11

SWEDEN

Nordisk Elektronik AB
Torshamnsgatan 39
Box 36
164 93 Kista
Tel: (46) 08-03 46 30
TLX: 105 47

SWITZERLAND

Industrade A.G.
Hertistrasse 31
8304 Wallisellen
Tel: (41) (01) 8328111
TLX: 56788

TURKEY

EMPA Electronic
Lindwurmstrasse 95A
8000 Muenchen 2
Tel: (49) 089/53 80 570
TLX: 528573

UNITED KINGDOM

Accent Electronic Components Ltd.
Jubilee House, Jubilee Road
Letchworth, Herts SG6 1QH
Tel: (44) (0462) 670011
FAX: (44) (0462) 682467
TWX: 826505

Bytech Components Ltd.
12A Cedarwood
Chineham Business Park
Crockford Lane
Basingstoke
Hants RG24 0WD
Tel: (0256) 707107
FAX: 0256-707162

Conformix
Unit 5
A1M Business Centre
Dixons Hill Road
Welham Green
South Hatfield
Herts AL9 7JE
Tel: (07072) 73282
FAX: (07072) 61678

Bytech Systems
3 The Western Centre
Western Road
Bracknell RG12 1RW
Tel: (44) (0344) 55333
FAX: (44) (0344) 867270
TWX: 849624

Jermyn
Vestry Estate
Otford Road
Sevenoaks
Kent TN14 5EU
Tel: (44) (0732) 450144
FAX: (44) (0732) 451251
TWX: 95142

MMD Ltd.
3 Bennet Court
Bennet Road
Reading
Berkshire RG2 0QX
Tel: (44) (0734) 313232
FAX: (44) (0734) 313255
TWX: 846669

Rapid Recall, Ltd.
Rapid House
Oxford Road
High Wycombe
Buckinghamshire HP11 2EE
Tel: (44) (0494) 26271
FAX: (44) (0494) 21860
TWX: 837931

Rapid Recall, Ltd.
28 High Street
Nantwich
Cheshire CW5 5AS
Tel: (0270) 627505
FAX: (0270) 629883
TWX: 36329

WEST GERMANY

Electronic 2000 AG
Stahlgruberring 12
8000 Muenchen 82
Tel: (49) 089/42001-0
TLX: 522561

ITT Multikomponent GmbH
Postfach 1265
Bahnhofstrasse 44
7141 Moeglingen
Tel: (49) 07141/4879
TLX: 7264472

Jermyn GmbH
Im Dachsstueck 9
6250 Limburg
Tel: (49) 06431/508-0
TLX: 415257-0

Metrologie GmbH
Meglingerstrasse 49
8000 Muenchen 71
Tel: (49) 089/78042-0
TLX: 5213189

Proelectron Vertriebs GmbH
Max Planck Strasse 1-3
6072 Dreieich
Tel: (49) 06103/30434-3
TLX: 417903

YUGOSLAVIA

H.R. Microelectronics Corp.
2005 de la Cruz Blvd., Ste. 223
Santa Clara, CA 95050
U.S.A.
Tel: (1) (408) 988-0286
TLX: 387452

Rapido Electronic Components S.p.a.
Via C. Beccaria, 8
34133 Trieste
Italia
Tel: (39) 040/360555
TLX: 460461

INTERNATIONAL SALES OFFICES

AUSTRALIA

Intel Australia Pty. Ltd.
Unit 13
Allamble Grove Business Park
25 Frenchs Forest Road East
Frenchs Forest, NSW, 2086
Tel: 61-2975-3300
FAX: 61-2975-3375

BRAZIL

Intel Semicondutores do Brazil LTDA
Av. Paulista, 1159-CJS 404/405
01311 - Sao Paulo - S.P.
Tel: 55-11-287-5899
TLX: 3911153146 ISDB
FAX: 55-11-287-5119

CHINA/HONG KONG

Intel PRC Corporation
15/F, Office 1, Citic Bldg.
Jian Guo Men Wai Street
Beijing, PRC
Tel: (1) 500-4850
TLX: 22947 INTEL CN
FAX: (1) 500-2953

Intel Semiconductor Ltd.*
10/F East Tower
Bond Center
Queensway, Central
Hong Kong
Tel: (852) 844-4555
FAX: (852) 868-1989

INDIA

Intel Asia Electronics, Inc.
4/2, Samrah Plaza
St. Mark's Road
Bangalore 560001
Tel: 011-91-812-215065
TLX: 953-845-2646 INTL IN
FAX: 091-812-215067

JAPAN

Intel Japan K.K.
5-6 Tokodai, Tsukuba-shi
Ibaraki, 300-26
Tel: 0298-47-8511
TLX: 3656-160
FAX: 0298-47-8450

Intel Japan K.K.*
Daiichi Mitsugi Bldg.
1-8889 Fuchu-cho
Fuchu-shi, Tokyo 183
Tel: 0423-60-7871
FAX: 0423-60-0315

Intel Japan K.K.*
Bldg. Kumagaya
2-69 Hon-cho
Kumagaya-shi, Saitama 360
Tel: 0485-24-6871
FAX: 0485-24-7518

Intel Japan K.K.*
Kawa-asa Bldg.
2-11-5 Shin-Yokohama
Kohoku-ku, Yokohama-shi
Kanagawa, 222
Tel: 045-474-7661
FAX: 045-471-4394

Intel Japan K.K.*
Ryokuchi-Eki Bldg.
2-4-1 Terauchi
Toyonaka-shi, Osaka 560
Tel: 06-863-1091
FAX: 06-863-1084

Intel Japan K.K.
Shinmaru Bldg.
1-5-1 Marunouchi
Chiyoda-ku, Tokyo 100
Tel: 03-201-3621
FAX: 03-201-6850

Intel Japan K.K.
Green Bldg.
1-16-20 Nishiki
Naka-ku, Nagoya-shi
Aichi 450
Tel: 052-204-1261
FAX: 052-204-1285

KOREA

Intel Korea, Ltd.
16th Floor, Life Bldg.
61 Yoido-dong, Youngdeungpo-Ku
Seoul 150-010
Tel: (2) 784-8186, 8286, 8386
TLX: K29312 INTELKO
FAX: (2) 784-8096

SINGAPORE

Intel Singapore Technology, Ltd.
101 Thomson Road #21-05/06
United Square
Singapore 1130
Tel: 250-7811
TLX: 39921 INTEL
FAX: 250-9256

TAIWAN

Intel Technology Far East Ltd.
8th Floor, No. 205
Bank Tower Bldg.
Tung Hua N. Road
Taipei
Tel: 886-2-716-9660
FAX: 886-2-717-2455

INTERNATIONAL DISTRIBUTORS/REPRESENTATIVES

ARGENTINA

Dafsys S.R.L.
Chacabuco, 90-6 Piso
1069-Buenos Aires
Tel: 54-1-334-7726
FAX: 54-1-334-1871

AUSTRALIA

Email Electronics
15-17 Hume Street
Huntingdale, 3166
Tel: 011-61-3-544-8244
TLX: AA 30895
FAX: 011-61-3-543-8179

NSD-Australia
205 Middleborough Rd.
Box Hill, Victoria 3128
Tel: 03 8900970
FAX: 03 8990819

BRAZIL

Elebra Componentes
Rua Geraldo Flausina Gomes, 78
7 Andar
04575 - Sao Paulo - S.P.
Tel: 55-11-534-9641
TLX: 55-11-54593/54591
FAX: 55-11-534-9424

CHINA/HONG KONG

Novel Precision Machinery Co., Ltd.
Room 728 Trade Square
681 Cheung Sha Wan Road
Kowloon, Hong Kong
Tel: (852) 360-8999
TWX: 32032 NVTNL HX
FAX: (852) 725-3695

INDIA

Micronic Devices
Arun Complex
No. 65 D.V.G. Road
Basavanagudi
Bangalore 560 004
Tel: 011-91-812-600-631
011-91-812-611-365
TLX: 9538458332 MDBG

Micronic Devices
No. 516 5th Floor
Swastik Chambers
Sion, Trombay Road
Chembur
Bombay 400 071
TLX: 9531 171447 MDEV

Micronic Devices
25/8, 1st Floor
Bada Bazaar Marg
Old Rajinder Nagar
New Delhi 110 060
Tel: 011-91-11-5723509
011-91-11-589771
TLX: 031-63253 MDND IN

Micronic Devices
6-3-348/12A Dwarakapuri Colony
Hyderabad 500 482
Tel: 011-91-842-226748

S&S Corporation
1587 Kooser Road
San Jose, CA 95118
Tel: (408) 978-6216
TLX: 820281
FAX: (408) 978-8635

JAPAN

Asahi Electronics Co. Ltd.
KMM Bldg. 2-14-1 Asano
Kokurakita-ku
Kitakyushu-shi 802
Tel: 093-511-6471
FAX: 093-551-7861

CTC Components Systems Co., Ltd.
4-8-1 Dobashi, Miyamae-ku
Kawasaki-shi, Kanagawa 213
Tel: 044-852-5121
FAX: 044-877-4268

Dia Semicon Systems, Inc.
Flower Hill Shinmachi Higashi-kan
1-23-9 Shinmachi, Setagaya-ku
Tokyo 154
Tel: 03-439-1600
FAX: 03-439-1601

Okaya Koki
2-4-18 Sakae
Naka-ku, Nagoya-shi 460
Tel: 052-204-2916
FAX: 052-204-2901

Ryoyo Electro Corp.
Konwa Bldg.
1-12-22 Tsukiji
Chuo-ku, Tokyo 104
Tel: 03-546-5011
FAX: 03-546-5044

KOREA

J-Tek Corporation
Dong Sung Bldg. 9/F
158-24, Samsung-Dong, Kangnam-Ku
Seoul 135-090
Tel: (822) 557-8039
FAX: (822) 557-8304

Samsung Electronics
Samsung Main Bldg.
150 Taepyung-Ro-2KA, Chung-Ku
Seoul 100-102
C.P.O. Box 8780
Tel: (822) 751-3680
TWX: KORSST K 27970
FAX: (822) 753-9065

MEXICO

SSB Electronics, Inc.
675 Palomar Street, Bldg. 4, Suite A
Chula Vista, CA 92011
Tel: (619) 585-3253
TLX: 287751 CBALL UR
FAX: (619) 585-8322

Dicopel S.A.
Tochtli 368 Fracc. Ind. San Antonio
Azcapotzalco
C.P. 02760-Mexico, D.F.
Tel: 52-5-561-3211
TLX: 177 3790 Dicome
FAX: 52-5-561-1279

PHI S.A. de C.V.
Fco. Villa esq. Ajusco s/n
Cuernavaca – Morelos
Tel: 52-73-13-9412
FAX: 52-73-17-5333

NEW ZEALAND

Email Electronics
36 Olive Road
Penrose, Auckland
Tel: 011-64-9-591-155
FAX: 011-64-9-592-681

SINGAPORE

Electronic Resources Pte, Ltd.
17 Harvey Road
#03-01 Singapore 1336
Tel: (65) 283-0888
TWX: RS 56541 ERS
FAX: (65) 289-5327

SOUTH AFRICA

Electronic Building Elements
178 Erasmus St. (off Watermeyet St.)
Meyerspark, Pretoria, 0184
Tel: 011-2712-803-7680
FAX: 011-2712-803-8294

TAIWAN

Micro Electronics Corporation
12th Floor, Section 3
285 Nanking East Road
Taipei, R.O.C.
Tel: (886) 2-7198419
FAX: (886) 2-7197916

Acer Sertek Inc.
15th Floor, Section 2
Chien Kuo North Rd.
Taipei 18479 R.O.C.
Tel: 886-2-501-0055
TWX: 23756 SERTEK
FAX: (886) 2-5012521

*Field Application Location

DOMESTIC SERVICE OFFICES

ALASKA

Intel Corp.
c/o TransAlaska Network
1515 Lore Rd.
Anchorage 99507
Tel: (907) 522-1776

Intel Corp.
c/o TransAlaska Data Systems
c/o GCI Operations
520 Fifth Ave., Suite 407
Fairbanks 99701
Tel: (907) 452-6264

ARIZONA

*Intel Corp.
410 North 44th Street
Suite 500
Phoenix 85008
Tel: (602) 231-0386
FAX: (602) 244-0446

*Intel Corp.
500 E. Fry Blvd., Suite M-15
Sierra Vista 85635
Tel: (602) 459-5010

ARKANSAS

Intel Corp.
c/o Federal Express
1500 West Park Drive
Little Rock 72204

CALIFORNIA

*Intel Corp.
21515 Vanowen St., Ste. 116
Canoga Park 91303
Tel: (818) 704-8500

*Intel Corp.
300 N. Continental Blvd.
Suite 100
El Segundo 90245
Tel: (213) 640-6040

*Intel Corp.
1900 Prairie City Rd.
Folsom 95630-9597
Tel: (916) 351-6143

*Intel Corp.
9665 Chesapeake Dr., Suite 325
San Diego 92123
Tel: (619) 292-8086

**Intel Corp.
400 N. Tustin Avenue
Suite 450
Santa Ana 92705
Tel: (714) 835-9642

**Intel Corp.
2700 San Tomas Exp., 1st Floor
Santa Clara 95051
Tel: (408) 970-1747

COLORADO

*Intel Corp.
600 S. Cherry St., Suite 700
Denver 80222
Tel: (303) 321-8086

CONNECTICUT

*Intel Corp.
301 Lee Farm Corporate Park
83 Wooster Heights Rd.
Danbury 06811
Tel: (203) 748-3130

FLORIDA

**Intel Corp.
800 Fairway Dr., Suite 160
Deerfield Beach 33441
Tel: (305) 421-0506
FAX: (305) 421-2444

*Intel Corp.
5850 T.G. Lee Blvd., Ste. 340
Orlando 32822
Tel: (407) 240-8000

GEORGIA

*Intel Corp.
20 Technology Park, Suite 150
Norcross 30092
Tel: (404) 449-0541

5523 Theresa Street
Columbus 31907

HAWAII

**Intel Corp.
Honolulu 96820
Tel: (808) 847-6738

ILLINOIS

**†Intel Corp.
Woodfield Corp. Center III
300 N. Martingale Rd., Ste. 400
Schaumburg 60173
Tel: (708) 605-8031

INDIANA

*Intel Corp.
8910 Purdue Rd., Ste. 350
Indianapolis 46268
Tel: (317) 875-0623

KANSAS

*Intel Corp.
10985 Cody, Suite 140
Overland Park 66210
Tel: (913) 345-2727

KENTUCKY

Intel Corp.
133 Walton Ave., Office 1A
Lexington 40508
Tel: (606) 255-2957

Intel Corp.
896 Hillcrest Road, Apt. A
Radcliff 40160 (Louisville)

LOUISIANA

Hammond 70401
(serviced from Jackson, MS)

MARYLAND

**Intel Corp.
10010 Junction Dr., Suite 200
Annapolis Junction 20701
Tel: (301) 206-2860

MASSACHUSETTS

**Intel Corp.
Westford Corp. Center
3 Carlisle Rd., 2nd Floor
Westford 01886
Tel: (508) 692-0960

MICHIGAN

*Intel Corp.
7071 Orchard Lake Rd., Ste. 100
West Bloomfield 48322
Tel: (313) 851-8905

MINNESOTA

*Intel Corp.
3500 W. 80th St., Suite 360
Bloomington 55431
Tel: (612) 835-6722

MISSISSIPPI

Intel Corp.
c/o Compu-Care
2001 Airport Road, Suite 205F
Jackson 39208
Tel: (601) 932-6275

MISSOURI

*Intel Corp.
4203 Earth City Exp., Ste. 131
Earth City 63045
Tel: (314) 291-1990

Intel Corp.
Route 2, Box 221
Smithville 64089
Tel: (913) 345-2727

NEW JERSEY

**Intel Corp.
300 Sylvan Avenue
Englewood Cliffs 07632
Tel: (201) 567-0821

*Intel Corp.
Parkway 109 Office Center
328 Newman Springs Road
Red Bank 07701
Tel: (201) 747-2233

NEW MEXICO

Intel Corp.
Rio Rancho 1
4100 Sara Road
Rio Rancho 87124-1025
(near Albuquerque)
Tel: (505) 893-7000

NEW YORK

*Intel Corp.
2950 Expressway Dr. South
Suite 130
Islandia 11722
Tel: (516) 231-3300

Intel Corp.
Westage Business Center
Bldg. 300, Route 9
Fishkill 12524
Tel: (914) 897-3860

Intel Corp.
5858 East Molloy Road
Syracuse 13211
Tel: (315) 454-0576

NORTH CAROLINA

*Intel Corp.
5800 Executive Center Drive
Suite 105
Charlotte 28212
Tel: (704) 568-8966

**Intel Corp.
5540 Centerview Dr., Suite 215
Raleigh 27606
Tel: (919) 851-9537

OHIO

**Intel Corp.
3401 Park Center Dr., Ste. 220
Dayton 45414
Tel: (513) 890-5350

*Intel Corp.
25700 Science Park Dr., Ste. 100
Beachwood 44122
Tel: (216) 464-2736

OREGON

**Intel Corp.
15254 N.W. Greenbrier Parkway
Building B
Beaverton 97005
Tel: (503) 645-8051

PENNSYLVANIA

*†Intel Corp.
925 Harvest Drive
Suite 200
Blue Bell 19422
Tel: (215) 641-1000
1-800-468-3548
FAX: (215) 641-0785

**†Intel Corp.
400 Penn Center Blvd., Ste. 610
Pittsburgh 15235
Tel: (412) 823-4970

*Intel Corp.
1513 Cedar Cliff Dr.
Camp Hill 17011
Tel: (717) 761-0860

PUERTO RICO

Intel Corp.
South Industrial Park
P.O. Box 910
Las Piedras 00671
Tel: (809) 733-8616

TEXAS

**Intel Corp.
Westech 360, Suite 4230
8911 Capitol of Texas Hwy.
Austin 78752-1239
Tel : (512) 794-8086

**†Intel Corp.
12000 Ford Rd., Suite 401
Dallas 75234
Tel: (214) 241-8087

**Intel Corp.
7322 SW Freeway, Suite 1490
Houston 77074
Tel: (713) 988-8086

UTAH

Intel Corp.
428 East 6400 South
Suite 104
Murray 84107
Tel: (801) 263-8051
FAX: (801) 268-1457

VIRGINIA

*Intel Corp.
1504 Santa Rosa Rd., Ste. 108
Richmond 23288
Tel: (804) 282-5668

WASHINGTON

**Intel Corp.
155 108th Avenue N.E., Ste. 386
Bellevue 98004
Tel: (206) 453-8086

CANADA

ONTARIO

**Intel Semiconductor of Canada, Ltd.
2650 Queensview Dr., Ste. 250
Ottawa K2B 8H6
Tel: (613) 829-9714

**Intel Semiconductor of Canada, Ltd.
190 Attwell Dr., Ste. 102
Rexdale (Toronto) M9W 6H8
Tel: (416) 675-2105

QUEBEC

**Intel Semiconductor of Canada, Ltd.
1 Rue Holiday
Suite 115
Tour East
Pt. Claire H9R 5N3
Tel: (514) 694-9130
FAX: 514-694-0064

CUSTOMER TRAINING CENTERS

CALIFORNIA

2700 San Tomas Expressway
Santa Clara 95051
Tel: 1-800-328-0386

MARYLAND

10010 Junction Dr.
Suite 200
Annapolis Junction 20701
Tel: 1-800-328-0386

SYSTEMS ENGINEERING OFFICES

MINNESOTA

3500 W. 80th Street
Suite 360
Bloomington 55431
Tel: (612) 835-6722

NEW YORK

2950 Expressway Dr., South
Islandia 11722
Tel: (506) 231-3300

*Carry-in locations
**Carry-in/mail-in locations

UNITED STATES
Intel Corporation
3065 Bowers Avenue
Santa Clara, CA 95051

JAPAN
Intel Japan K.K.
5-6 Tokodai, Tsukuba-shi
Ibaraki, 300-26

FRANCE
Intel Corporation S.A.R.L.
1, Rue Edison, BP 303
78054 Saint-Quentin-en-Yvelines Cedex

UNITED KINGDOM
Intel Corporation (U.K.) Ltd.
Pipers Way
Swindon
Wiltshire, England SN3 1RJ

GERMANY
Intel GmbH
Dornacher Strasse 1
8016 Feldkirchen bei Muenchen

HONG KONG
Intel Semiconductor Ltd.
10/F East Tower
Bond Center
Queensway, Central

CANADA
Intel Semiconductor of Canada, Ltd.
190 Attwell Drive, Suite 500
Rexdale, Ontario M9W 6H8

intel®

Programmable Logic

System designers select programmable logic devices to develop smaller systems, reduce system costs, obtain higher performance and reliability, maintain design security and increase flexibility.

The burgeoning market of programmable logic has brought a plethora of terms and technologies into the industry. With so many choices, which is the right device to select?

Intel's programmable logic devices are based on our erasable programmable read-only memory technology for instant reconfigurability. All Intel programmable logic devices (PLDs) offer you the ability to customize your system logic by tailoring the devices to meet your performance requirements.

With our new microcomputer programmable logic device (μPLD) family, Intel marks a technology breakthrough by offering CMOS devices in speeds faster than bipolar technology. Design engineers can now use low-power CMOS devices tuned to the microprocessors for developing leading-edge high-performance systems.

This handbook contains data sheets, application notes and technical briefs for Intel's PLD family, including information on the PC-based tool, iPLDS II, used to design, compile and program Intel's PLDs.

Printed in USA/0191/45K/RRD/LK
Programmable Logic

ISBN# 1-55512-124-1