Final

Advanced Micro Devices

PALCE16V8H-15/25

EE CMOS Universal Programmable Array Logic

DISTINCTIVE CHARACTERISTICS

- **Pin, function and fuse-map compatible with all 20-pin GAL® devices**
- **Electrically erasable CMOS technology provides reconfigurable logic and full testability**
- **High speed CMOS technology**
  - 15-ns propagation delay for "-15" version
  - 25-ns propagation delay for "-25" version
- **Direct plug-in replacement for the PAL16R8 series and most of the PAL10H8 series**
- **Outputs programmable as registered or combinatorial in any combination**
- **Programmable output polarity**
- **Programmable enable/disable control**
- **Preloadable output registers for testability**
- **Automatic register reset on power up**
- **Cost-effective 20-pin plastic DIP and PLCC packages**
- **Programmable on standard device programmers**
- **Supported by PALASM® software**
- **Fully tested for high programming and functional yields and high reliability**

GENERAL DESCRIPTION

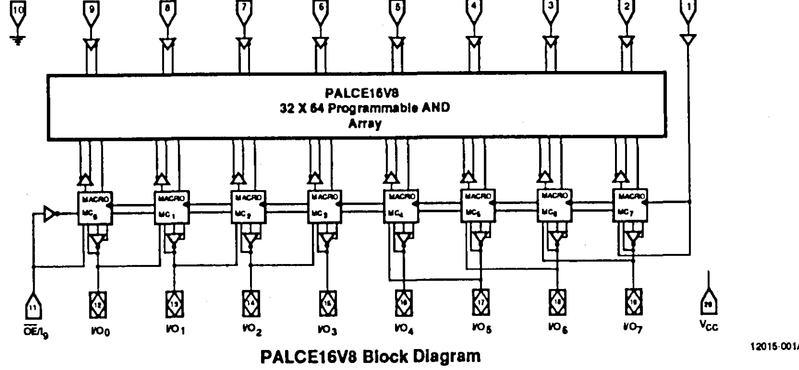
The PALCE16V8 is an advanced PAL® device built with low-power, high-speed, electrically-erasable CMOS technology. It is functionally compatible with all 20-pin GAL devices. The macrocells provide a universal device architecture. The PALCE16V8 will directly replace the PAL16R8 and PAL10H8 series devices, with the exception of the PAL16C1.

Device logic is automatically configured according to the user's design specification. Design is simplified by PALASM design software, allowing automatic creation of a programming file based on Boolean or state equations. PALASM software also verifies the design and can provide test vectors for the finished device. Programming can be accomplished on standard PAL device programmers.

The PALCE16V8 utilizes the familiar sum-of-products (AND/OR) architecture that allows users to implement complex logic functions easily and efficiently. Multiple levels of combinatorial logic can always be reduced to sum-of-products form, taking advantage of the very wide input gates available in PAL devices. The equations are programmed into the device through floating-gate cells in the AND logic array that can be erased electrically.

The fixed OR array allows up to eight data product terms per output for logic functions. The sum of these products feeds the output macrocell. Each macrocell can be programmed as registered or combinatorial with an active-HIGH or active-LOW output. The output configuration is determined by two global bits and one local bit controlling four multiplexers in each macrocell.

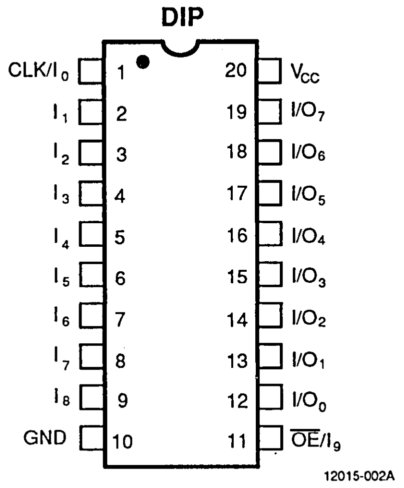
BLOCK DIAGRAM

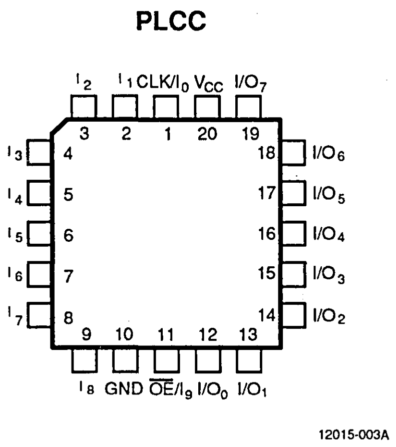


PALCE16V8 Block Diagram

Publication # 12015 Rev. A Amendment
Issue Date: April 1989

CONNECTION DIAGRAMS





Pin Designations:

I	=	Input
I/O	=	InputOutput
$\overline{OE}$	=	Output Enable
CLK	=	Clock
V_{CC}	=	Supply Voltage
GND	=	Ground

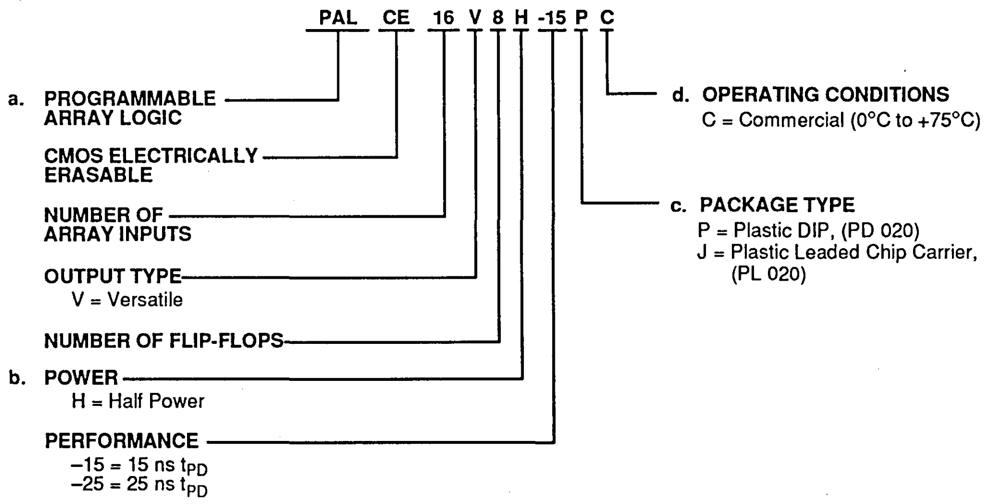
Note: Pin 1 is marked for orientation

ORDERING INFORMATION

Standard Products

AMD/MMI standard products are available in several packages. The order number (Valid Combination) is formed by a combination of:

a **Device Number**
b. **Speed/Power Option**
c. **Package Type**
d. **Operating Conditions**



a. **PROGRAMMABLE ARRAY LOGIC**

CMOS ELECTRICALLY ERASABLE

NUMBER OF ARRAY INPUTS

OUTPUT TYPE
V = Versatile

NUMBER OF FLIP-FLOPS

b. **POWER**
H = Half Power

PERFORMANCE
–15 = 15 ns t_{PD}
–25 = 25 ns t_{PD}

d. **OPERATING CONDITIONS**
C = Commercial (0°C to +75°C)

c. **PACKAGE TYPE**
P = Plastic DIP, (PD 020)
J = Plastic Leaded Chip Carrier, (PL 020)

Valid Combinations	
PALCE16V8H-15	PC, JC
PALCE16V8H-25	

Valid Combinations

The valid Combinations table lists configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

PIN DESCRIPTION

Symbol	Type	Function
V_{CC}		Five Volt Power Input.
GND		Ground
CLK/I_0	TTL input	Clock. If the CLK function is not used, it can used as a TTL input signal
$\overline{OE}$/I_9	TTL Input	Output Enable. If the $\overline{OE}$ function is not used, it can be used as a TTL input signal.
I_1..I_8	TTL inputs	Input 1 through Input 8
I/O_0..I/O_7	TTL I/O	I/O_0 through I/O_7

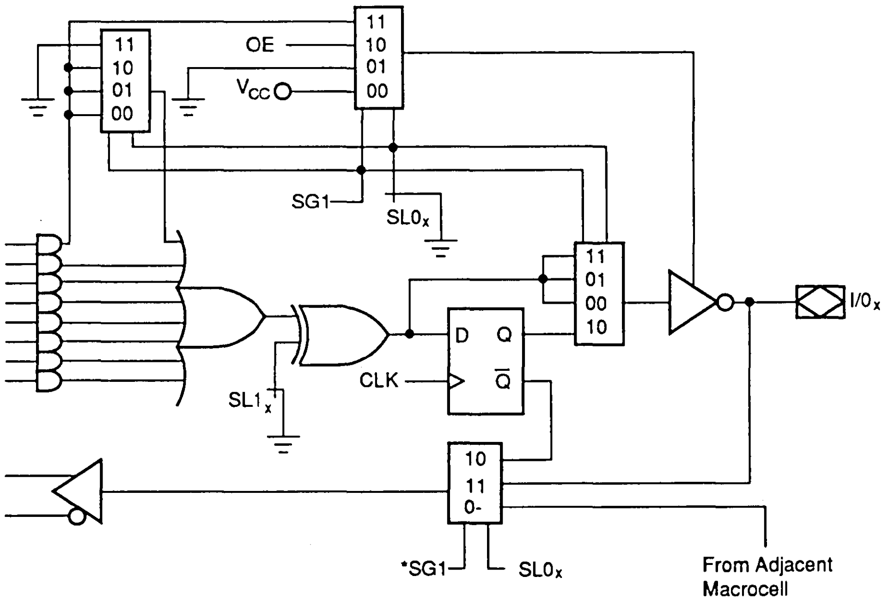
FUNCTIONAL DESCRIPTION

The PALCE16V8 is a universal PAL device. It has eight independently configurable macrocells (MC_0.. MC_7). The macrocells can be configured as registered output, combinatorial output, combinatorial I/O or dedicated input. The programming matrix implements a programmable AND logic array, which drives a fixed OR logic array. Buffers for device inputs have complementary outputs to provide user-programmable input signal polarity. Pins 1 and 11 serve either as array inputs or as clock (CLK) and output enable ($\overline{OE}$) for all flip-flops.

Unused input pins should be tied directly to VCC or GND. Product terms with all bits unprogrammed (disconnected) assume the logical HIGH state and product terms with both true and complement of any input signal connected assume a logical LOW state.

The programmable functions on the PALCE16V8 are automatically configured from the user's design specification, which can be in a number of formats. The design specification is processed by development software to verify the design and create a programming file. This file, once downloaded to a programmer, configures the device according to the user's desired function.

The user is given two design options with the PALCE16V8. First, it can be programmed as a standard PAL device from the PAL16R8 and PAL10H8 series. The PAL programmer manufacturer will supply device codes for the standard PAL device architectures to be used with the PALCE16V8. The programmer will program the PALCE16V8 in the corresponding architecture. This allows the user to use existing standard PAL device files without making any changes to them. This includes JEDEC files. Alternatively, the device can be programmed as a PALCE16V8. Here the user must use the PALCE16V8 device code. This option allows full utilization of the macrocell.



*In macrocells MC_0 and MC_7, SG1 is replaced by $\overline{SG0}$ on the feedback multiplexer.

12015-004A

PALCE16V8 Macrocell

Configuration Options

Each macrocell can be configured as one of the following: registered output, combinatorial output or dedicated input. In the registered output configuration, the output buffer is enabled by the $\overline{OE}$ pin. In the combinatorial configuration, the buffer is either controlled by a product term or always enabled. In the dedicated input configuration, it is always disabled. With the exception of MC_0 and MC_7, a macrocell configured as a dedicated input derives the input signal from an adjacent I/O. MC_0 derives its input from pin 11($\overline{OE}$) and MC_7 from pin 1 (CLK).

The macrocell configurations are controlled by bits stored in the configuration control word. It contains 2 global bits (SG0 and SG1) and 16 local bits ($SL0_0$ through $SL0_7$ and $SL1_0$ through $SL1_7$). SG0 determines whether registers will be allowed. SG1 determines whether the PALCE16V8 will emulate a 16R8 family or a PAL10H8 family device. $SL0_x$, in conjunction with SG1, selects the configuration of the macrocell, and $SL1_x$ sets the output as either active LOW or active HIGH for the individual macrocell.

The configuration bits work by acting as control inputs for the multiplexers in the macrocell. There are four multiplexers: a product term input, an enable select, an output select, and a feedback select multiplexer. SG1 and $SL0_x$ are the control signals for all four multiplexers. In addition, SL0 for the adjacent I/O is a control input to the feedback multiplexer. In MC_0 and MC_7, $\overline{SG0}$ replaces SG1 on the feedback multiplexer. This accommodates CLK being the adjacent pin for MC_7 and $\overline{OE}$ for MC_0.

Registered Output Configuration

The control bit settings are SG0 = 0, SG1 = 1 and $SL0_x$ = 0. There is only one registered configuration. All eight product terms are available as inputs to the OR gate. Data polarity is determined by $SL1_x$. The flip-flop is loaded on the LOW to HIGH transition of CLK. The feedback path is from $\overline{Q}$ on the register. The output buffer is enabled by $\overline{OE}$.

Combinatorial Configurations

The PALCE16V8 has three combinatorial output configurations: dedicated output in a non-registered device, I/O in a non-registered device and I/O in a registered device.

Dedicated Output In a Non-Registered-Device

The control bit settings are SG0 = 1, SG1 = 0 and $SL0_x$ = 0. All eight product terms are available to the OR gate. Because the macrocell is a dedicated output, the feedback is not used. Because CLK and $\overline{OE}$ are not used in a non-registered device, pins 1 and 11 are available as input signals. Pin 1 will use the feedback path of MC_7 and pin 11 will use the feedback path of MC_0.

Combinatorial I/O In a Non-Registered Device

The control settings are SG0 = 1, SG1 = 1, and $SL0_x$ = 1. Only seven product terms are available to the OR gate. The eighth product term is used to enable the output buffer. The signal at the I/O pin is fed back to the AND array via the feedback multiplexer. This allows the pin to be used as an input.

Because CLK and $\overline{OE}$ are not used in a non-registered device, pins 1 and 11 are available as inputs. Pin 1 will use the feedback path of MC_7 and pin 11 will use the feedback path of MC_0.

Combinatorial I/O in a Registered Device

The control bit settings are SG0 = 0, SG1 = 1 and $SL0_x$ = 1. Only seven product terms are available to the OR gate. The eighth product term is used as the output enable. The feedback signal is the corresponding I/O signal.

Dedicated Input Configuration

The control bit settings are SG0 = 1, SG1 = 0 and $SL0_x$ = 1. The output buffer is disabled. Except for MC_0 and MC_7 the feedback signal is an adjacent I/O. For MC_0 and MC_7 the feedback signals are pins 1 and 11. These configurations are summarized in Table 1 and illustrated in Figure 2.

Macrocell Configuration				
SG0	**SG1**	**SL0x**	**Cell Configuration**	**Devices Emulated**
Device Uses Registers				
0	1	0	Registered Output	PAL 16R8, 16R6, 16R4
0	1	1	Combinatorial I/O	PAL16R6, 16R4
Device Uses No Registers				
1	0	0	Combinatorial Output	PAL10H8, 12H6, 14H4, 16H2, 10L8, 12L6, 14L4, 16L2
1	0	1	Input	PAL12H6, 14H4, 16H2, 12L6, 14L4, 16L2
1	1	1	Combinatorial I/O	PAL16L8

Programmable Output Polarity

The polarity of each macrocell can be active HIGH or active LOW, either to match output signal needs or to reduce product terms. Programmable polarity allows Boolean expressions to be written in their most compact form (true or inverted), and the output can still be of the desired polarity. It can also save "DeMorganizing" efforts.

Selection is through a programmable bit $SL1_x$ which controls an exclusive-OR Gate at the output of the AND/OR logic. The output is active-HIGH if $SL1_x$ is "1" and active-LOW if $SL1_x$ is "0".

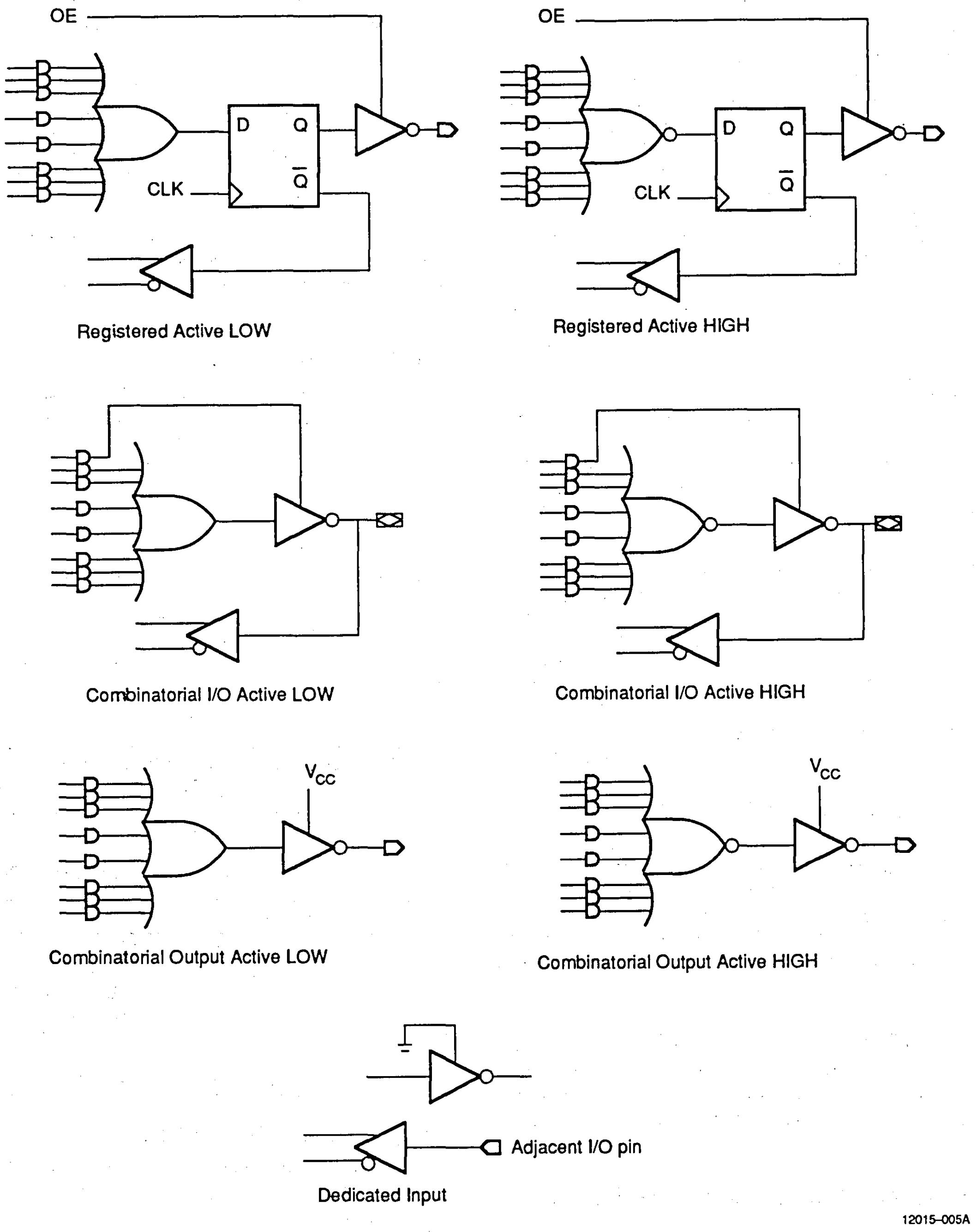

OE
CLK
D
Q
Q̄
Registered Active LOW
OE
CLK
D
Q
Q̄
Registered Active HIGH
Combinatorial I/O Active LOW
Combinatorial I/O Active HIGH
V_{CC}
Combinatorial Output Active LOW
V_{CC}
Combinatorial Output Active HIGH
Adjacent I/O pin
Dedicated Input
12015–005A

Figure 2. Macrocell Configurations

Power-Up Reset

All flip-flops power up to a logic LOW for predictable system initialization. Outputs of the PALCE16V8 will depend on whether they are selected as registered or combinatorial. If registered is selected, the output will be LOW. If combinatorial is selected, the output is a function of the logic.

Electronic Signature Word

An electronic signature word is provided in the PALCE16V8 device. It consists of 64 bits of programmable memory that can contain user-defined data. The signature data is always available to the user independent of the security bit.

Programming and Erasing

The PALCE16V8 can be programmed on standard logic programmers. Approved programmers are listed in this data sheet.

The PALCE16V8 may be erased to reset a previously configured device back to its virgin state. Bulk erase is automatically performed by the programming hardware. No special erase operation is required.

Security Bit

A security bit is provided on the PALCE16V8 as a deterrent to unauthorized copying of the array configuration patterns. Once programmed, this bit defeats readback of the programmed pattern by a device programmer, securing proprietary designs from competitors. However, programming and verification are also defeated by the security bit. The bit can only be erased in conjunction with the array during a bulk erase cycle.

Basic PAL Device Notation

The multi-input gates in the PAL device's programmable AND gate array are simplified in the logic diagrams. The PAL device notation for an AND gate, called a product term in a PAL device, is shown below.

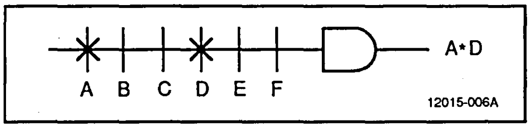


Figure 3. PAL Device AND Gate

This is equivalent to the standard logic notation below.

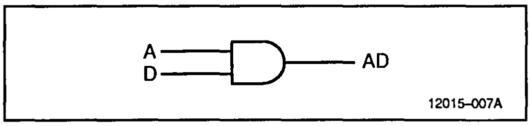


Figure 4. Standard AND Gate

Each vertical line in the PAL device is a potential input to the AND gate. At each crosspoint is a programmable bit, which provides a potential connection in the programmed state. The Xs in the diagram indicate a connection at the crosspoint.

In electrically erasable devices the crosspoints are originally disconnected. They are either connected or left open during device programming.

Multiplexers in the PAL device logic diagrams use a simple notation for maximum clarity. A 2:1 multiplexer that selects X when the control is LOW and Y when the control is HIGH is shown below.

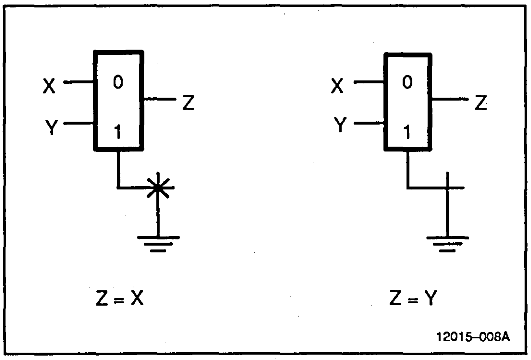
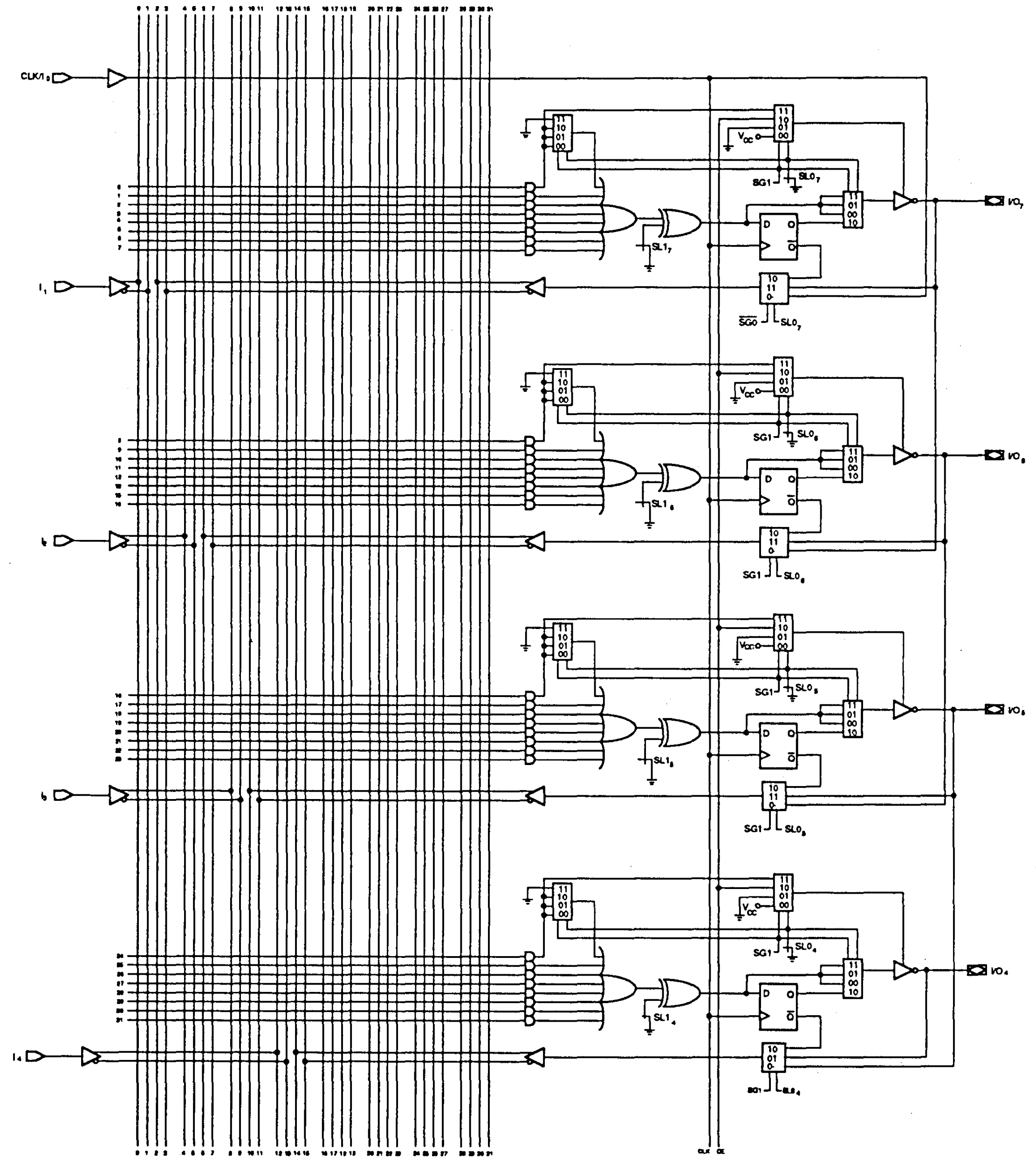
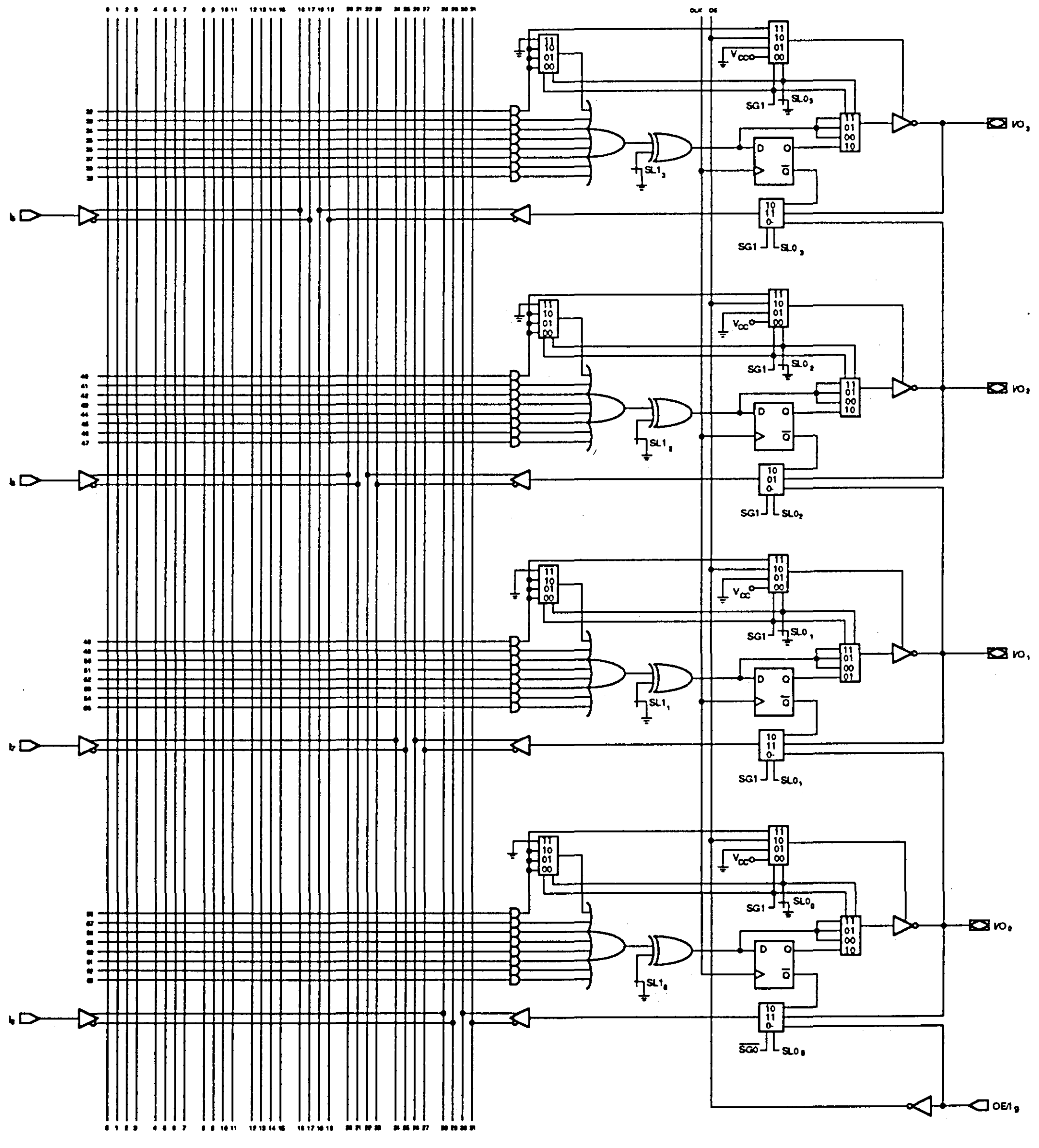


Figure 5. PAL Device Multiplexer

Notice that the control is operated by a programmable cell that is initially disconnected from GND, floating to Vcc, selecting the "1" path through the multiplexer. When the cell is programmed, it is connected to GND selecting the "0" path through the multiplexer.


CLK/I0
I1
I2
I3
I4
SL17
SL16
SL15
SL14
SG1
SL07
SL06
SL05
SL04
SG0
VCC
I/O7
I/O6
I/O5
I/O4
CLK
OE

12015-009A



12015-009A
Concluded

Figure 6. PALCE16V8 Logic Diagram

ABSOLUTE MAXIMUM RATINGS

Storage Temperature	–65°C to +150°C
Ambient Temperature under bias	–55°C to +125°C
Supply Voltage with Respect to Ground	–0.5V to +7.0V
DC Output Voltage	–0.5V to V_{CC} + 0.5V
DC Input Voltage	–0.5V to V_{CC} + 0.5V
Static Discharge Voltage	>2001V
Latchup Current (TA = 0°C to 75°C)	>100mA

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices

Temperature (T_A) Operating Free Air	0°C to +75°C
Supply Voltage (V_{CC})	+4.75V to +5.25V

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over operating range unless otherwise specified.

Parameter Symbol	Parameter Descriptions	Test Conditions	Min.	Max.	Unit
V_{OH}	Output HIGH Voltage	V_{CC} = MIN I_{OH} = –3.2 mA V_{IN} = V_{IH} or V_{IL}	2.4		V
V_{OL}	Output LOW Voltage	V_{CC} = MIN I_{OL} = 24mA V_{IN} = V_{IH} or V_{IL}		0.5	V
V_{IH}	Input HIGH Voltage Voltage for all Inputs (Note 1)	Guaranteed Input Logical HIGH	2.0		V
V_{IL}	Input LOW Voltage Voltage for all Inputs (Note 1)	Guaranteed Input Logical LOW		0.8	V
I_{IH}	Input Leakage Current	GND ≤ V_{IN} ≤ V_{CC} Max. (Note 2)		10	µA
I_{IL}				–10	
I_{OZH}	Off-State Output Current	GND ≥ V_{IN} ≥ V_{CC} Max. (Note 2)		10	µA
I_{OZL}				–10	
I_{OS}	Output Short-Circuit Current	V_{CC} = Max. V_{OUT} = 0V (Note 2)	–30	–130	mA
Icc	Supply Current	Outputs Open (I_O = 0A) V_{CC} = Max., F = 15MHz		90	mA

Notes:

1. These are absolute values with respect to device ground and all overshoots due to system or tester noise are included.
2. I/O pin leakage is the worst case of I_{IL} and I_{OZL} (or I_{IH} and I_{OZH}).
3. No more than one output should be shorted at a time and duration of the short-circuit should not exceed one second.

Capacitance (Note 1)

Parameter Symbol	Parameter Descriptions	Test Conditions	Typ.	Unit
C_{IN}	Input Capacitance	V_{CC} = 5.0V, T_A = +25°C	5	pF
C_{OUT}	Output Capacitance	V_{IN} = 2.0V at f = 1MHz	15	pF

Note:

1. These parameters are not 100% tested, but are evaluated at initial characterization and at any time the design is modified where capacitance may be affected.

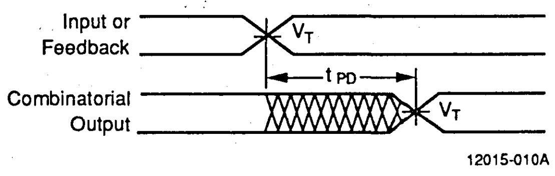
SWITCHING CHARACTERISTICS over Commercial operating range (Note 1)

Parameter Symbol	Parameter Description		–15 Min.	–15 Max.	–25 Min.	–25 Max.	Unit
t_{PD}	Input or Feedback to Combinatorial Output (Note 2)			15		25	ns
t_S	Setup Time from Input or Feedback to Clock		12		15		ns
t_H	Hold Time		0		0		ns
t_{CO}	Clock to Output			10		12	ns
t_{CF}	Clock to Feedback			8		10	ns
t_{WL}	Width of Clock	LOW	8		10		ns
t_{WH}		HIGH	8		10		
f_{MAX}	Maximum Frequency (Note 3)	External Feedback $1/(t_S+t_{CO})$	45.5		37		MHz
		Internal Feedback $1/(t_S+t_{CF})$	50.0		40		
		No Feedback $1/(t_{WH}+t_{WL})$	62.5		40		
t_{PZX}	$\overline{OE}$ to Output Enable (Note 4)			15		20	ns
t_{PXZ}	$\overline{OE}$ to Output Disable (Note 4)			15		20	ns
t_{EA}	Input to Output Enable (Notes 4 and 5)			15		25	ns
t_{ER}	Input to Output Disable (Notes 4 and 5)			15		25	ns

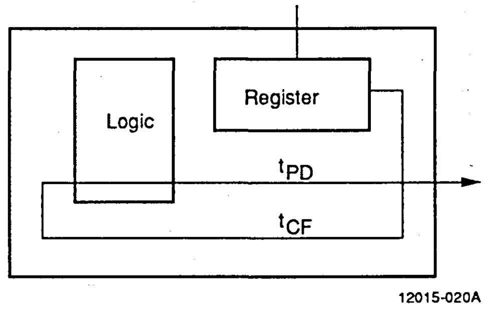
Notes:

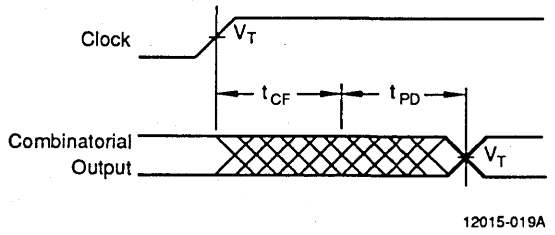
1. Commercial Test Conditions: $R_1 = 200\Omega$, $R_2 = 390\Omega$ (see switching test circuit).
2. t_{PD} is tested with S_1 closed and $C_L = 50pF$ (including jig capacitance). $V_{IH} = 3V$, $V_{IL} = 0V$, $V_{OH} = V_{OL} = 1.5V$.
3. These parameters are not 100% tested, but are calculated at initial characterization and at any time the design is modified where frequency may be affected.
4. For three-state outputs, enable times are tested with $C_L = 50pF$ to the 1.5V level; S_1 is open for high-impedance to HIGH tests and closed for high-impedance to LOW tests. Output disable times are tested with $C_L = 5pF$. HIGH to high-impedance tests are made to an output voltage of V_{OH} –0.5V with S_1 open; LOW to high-impedance tests are made to an output voltage of V_{OL} to +0.5V with S_1 closed.
5. Equivalent function to t_{PZX}, t_{PXZ} but using product term control.

SWITCHING WAVEFORMS

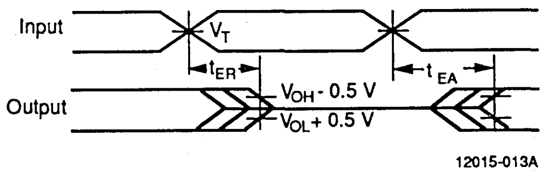


Combinatorial Output

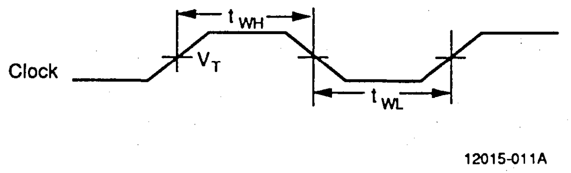




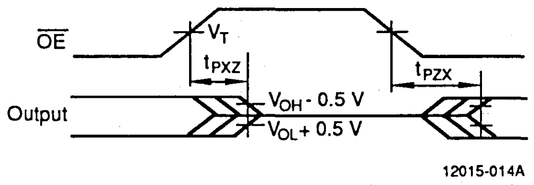
Clock to Feedback to Combinatorial Output (See Path at Right)



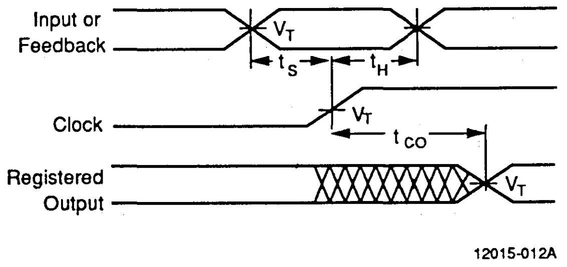
Input to Output Disable/Enable



Clock Width



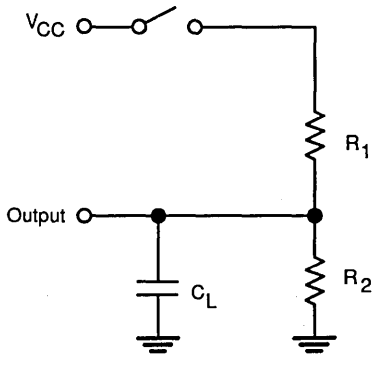
$\overline{OE}$ to Output Disable/Enable



Registered Output

Notes:
1. V_T = 1.5 V
2. Input pulse amplitude 0 V to 3.0 V
3. Input rise and fall times 2 – 5 ns typical

SWITCHING TEST CIRCUIT



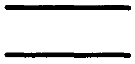
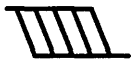
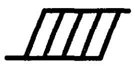
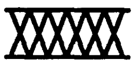
Switching Test Circuit

12015–019A

Notes on Testing Information

Specification	Switch S_1	C_L	R_1	R_2	Measured Output Value
t_{PD}, t_{CO}, t_{CF}	Closed	50 pF	200 Ω	390 Ω	1.5V
t_{PZX}, t_{EA}	Z->H: Open Z->L: Closed	50 pF	200 Ω	390 Ω	1.5V
t_{PXZ}, t_{ER}	H->Z: Open L->Z: Closed	50 pF	200 Ω	390 Ω	H->Z: V_{OH} –0.5V L->Z: V_{OL} +0.5V

Key to Switching Waveforms

WAVEFORM	INPUTS	OUTPUTS
	MUST BE STEADY	WILL BE STEADY
	MAY CHANGE FROM H TO L	WILL BE CHANGING FROM H TO L
	MAY CHANGE FROM L TO H	WILL BE CHANGING FROM L TO H
	DON'T CARE, ANY CHANGE PERMITTED	CHANGING, STATE UNKNOWN

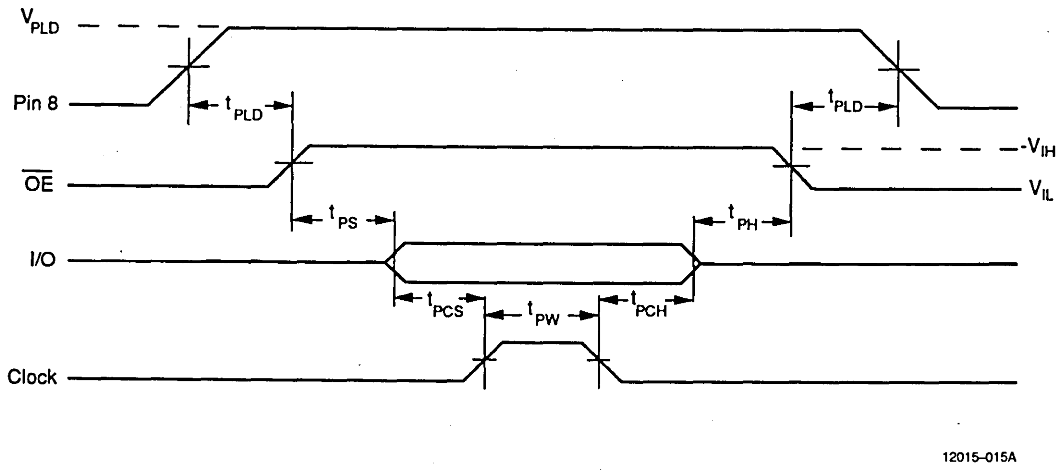
12015A–018A

Output Register Preload

The PRELOAD function allows the registers to be loaded from the output pins. This feature aids functional testing of sequential designs by allowing direct setting of output states. The procedure is as follows.

1. Raise V_{CC} to 5.0 V ± 0.5 V.
2. Set pin 8 to 10.0 V ± 0.5 V.
3. Set $\overline{OE}$ HIGH.
4. Apply the desired value (V_{IL}/V_{IH}) to all registered output pins. Leave combinatorial output pins floating.
5. Clock pin 1 from V_{IL} to V_{IH}.
6. Remove V_{IL}/V_{IH} from all registered outputs.
7. Lower pin 8 to V_{IL}/V_{IH}.
8. Enable the output registers by lowering $\overline{OE}$.
9. Verify for V_{OL}/V_{OH} at all registered output pins. Note that the output pin signal will be the inverse of the preload input.

Parameter Symbol	Parameter Description	Min.	Rec.	Max.	Unit
t_{PLD}	Setup and Hold Time from Preload (pin 8) to $\overline{OE}$	50	50		µs
t_{PS}	Setup Time from $\overline{OE}$ to Data	1	1		µs
t_{PH}	Hold Time from Data to $\overline{OE}$	1	1		µs
t_{PCS}	Setup Time from Data to Clock	1	1		µs
t_{PCH}	Hold Time from Clock to Data	1	1		µs
dV_r/dt	V_{PLD} Rising Slew Rate (pin 8)	10		100	V/µs
dV_f/dt	V_{PLD} Falling Slew Rate (pin 8)		2	3	V/µs



Preload Waveforms

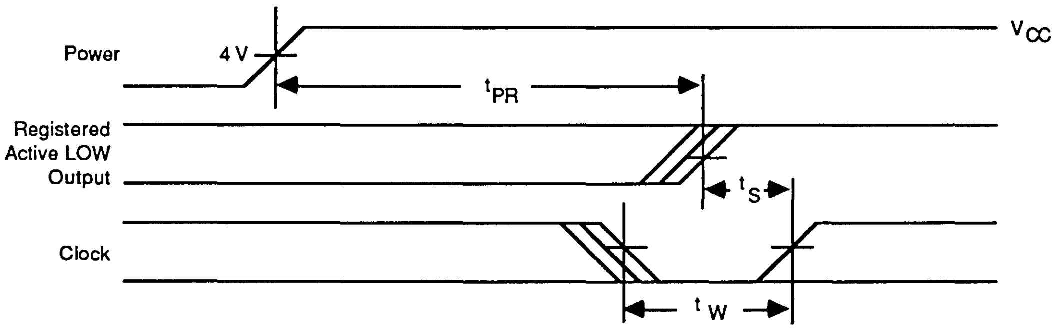
Power-Up Reset

The PALCE16V8 has been designed with the capability to reset during system power-up. Following power-up, all flip-flops will be reset to LOW. The output state will be HIGH independent of the logic polarity. This feature provides extra flexibility to the designer and is especially valuable in simplifying state machine initialization. A timing diagram and parameter table are shown below.

Due to the synchronous operation of the power-up reset and the wide range of ways Vcc can rise to its steady state, two conditions are required to insure a valid power-up reset. These conditions are:

1. The Vcc rise must be monotonic.
2. Following reset, the clock input must not be driven from LOW to HIGH until all applicable input and feedback setup times are met.

Parameter Symbol	Parameter Descriptions	Min.	Max.	Unit
t_{PR}	Power-Up Reset Time		100	µs
t_S	Input or Feedback Setup Time	See Switching Characteristics		
t_W	Clock width	See Switching Characteristics		



12015-017A

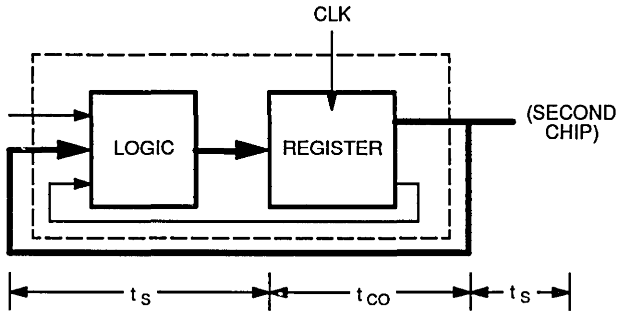
f_{MAX} Parameters

The parameter f_{MAX} is the maximum clock rate at which the device is guaranteed to operate. Because flexibility inherent in programmable logic devices offers a choice of clocked flip-flop designs, f_{MAX} is specified for three types of synchronous designs.

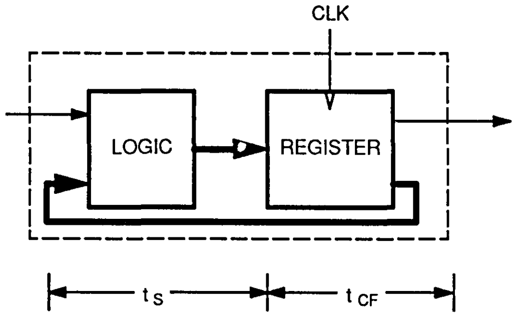
The first type of design is a state machine with feedback signals sent off-chip. This external feedback could go back to the device inputs, or to a second device in a multi-chip state machine. The slowest path defining the period is the sum of the clock-to-output time and the input setup time for the external signals ($t_S + t_{CO}$). The reciprocal, f_{MAX}, is the maximum frequency with external feedback or in conjunction with an equivalent speed device. This f_{MAX} is designated "f_{MAX} external."

The second type of design is a single-chip state machine with internal feedback only. In this case, flip-flop outputs are defined by the device inputs and flip-flop outputs. Under these conditions, the period is limited by the internal delay from the flip-flop outputs through the internal feedback and logic to the flip-flop inputs ($t_S + t_{CF}$). This fMAX is designated "f_{MAX} internal".

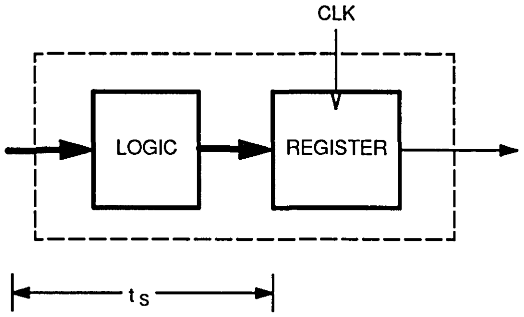
The third type of design is a simple data path application. In this case, input data is presented to the flip-flop and clocked through; no feedback is employed. Under these conditions, the period is limited by the sum of the data setup time and the data hold time ($t_S + t_H$). However, as lower limit for the period of each f_{MAX} type is the minimum clock period ($t_{WH} + t_{WL}$). Usually, this minimum clock period designates the period for the third f_{MAX}, designated "f_{MAX} no feedback".



f_{max} External Feedback; $1/(t_S + t_{CO})$



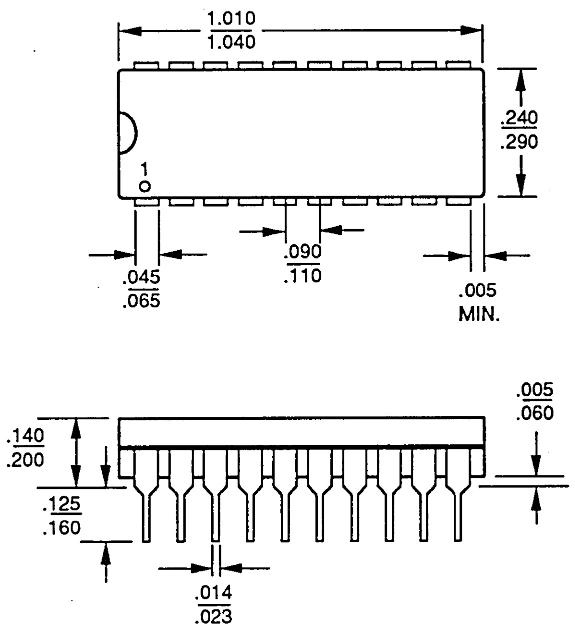
f_{max} Internal Feedback; $1/(t_S + t_{CF})$



f_{max} No Feedback; $1/(t_S + t_H)$ or $1/(t_{WH} + t_{WL})$

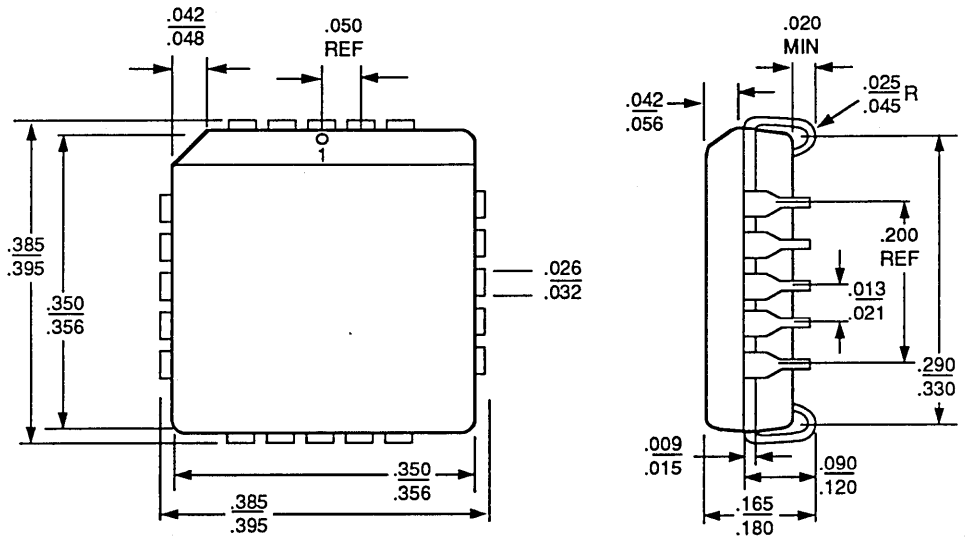
12015-020A

PHYSICAL DIMENSIONS
PD 020



12015-021A

PL 020



06970D

Programmers/Development Systems (Subject to change)

MANUFACTURER	PROGRAMMER CONFIGURATION	
Adams MacDonald 2999 Monterey/Salinas Hwy. Monterey, CA 93940 (408) 373-3607	Contact Manufacturer	
Data I/O Corporation Willow Road NE PO Box 97046 Redmond, WA 98073-9746 (800) 247-5700	System 29B LogicPak™ 303A-V04 Adapter 303A-011A/Rev. V10	UniSite Rev. 2.5 Family/Pinout Code 80-55
Digelec Inc. 1602 Lawrence Avenue, Suite 113 Ocean, NJ 07712 (201) 493-2420	Contact Manufacturer	
Kontron Electronics Inc.Contact Manufacturer 1230 Charleston Road Mountain View, CA 94039-7230 (415) 965-7020	Contact Manufacturer	
Logical Devices 1201 E. Northwest 65th Place Fort Lauderdale, FL 33309	Contact Manufacturer	
Micropross Parc d'Activite des Pres 5, rue Denis-Papin 59650 Villeneuve-d'Ascq (20) 47.90.40	Contact Manufacturer	
Stag Microsystems Inc. 1600 Wyatt Drive, Suite 3 Santa Clara, CA 95054 (408) 988-1118	Contact Manufacturer	
Varix Corporation 1210 E. Campbell Road, Suite 100 Richardson, TX 75081 (214) 437-0777	Contact Manufacturer	

MANUFACTURER	SOFTWARE DEVELOPMENT SYSTEM
Advanced Micro Devices 901 Thompson Place Sunnyvale, CA 94088-3453 (800) 222-9323	PALASM 2.23D
Data I/O Corporation 10525 Willow Road NE PO Box 97046 Redmond, WA 98073-9746 (800) 247-5700	Contact Manufacturer
Personal CAD Systems Assisted Technology Division 1290 Parkmoor Avenue San Jose, CA 95126 (408) 971-1300	Contact Manufacturer

PALASM 2 SOFTWARE SUPPORT FOR THE PALCE16V8

About this Section

This section describes PALASM 2 software special considerations for the PALCE16V8. It is intended as a supplement to the PALASM 2 software user documentation in part 4 of the 1988 PAL Device Data Book. If you do not already have the Data Book, contact your local AMD sales office for a copy.

Boolean Equation Design Entry

1. The pin list for the PALCE16V8 follows.

   ```
   ;1   2    3    4    5    6    7    8    9    10
   CLK  I2   I3   I4   I5   I6   I7   I8   I9   GND

   ;11  12   13   14   15   16   17   18   19   20
   OE   O1   O2   O3   O4   O5   O6   O7   O8   VCC
   ```

   Note: The lines beginning with a semicolon (;) are comments and are ignored by the software.

2. You use the SIGNATURE command to program the signature fuse. The command must be used in the Declaration segment of your design file. It must follow the CHIP statement. If you enter it in the Equations segment, the software displays an error message.

   The signature command syntax follows.

Syntax

```
SIGNATURE = number
            or
            string
```

Each of the syntax options is defined below.

Syntax option 1:

```
SIGNATURE = number
```

The number you use can be

• binary	#B or #b
• octal	#O or #o
• decimal	#D or #d
• hexadecimal	#H or #h

Notice that each number base is specified by an upper case or lower case designator. The examples below illustrate different ways to specify the signature as a number.

Examples

```
SIGNATURE = 123456

SIGNATURE = #D845

SIGNATURE = #H  1976A5
```

Note the following

- The space in the last example is allowed, but is deleted by the software.
- Number designators are optional. If you do not use a number designator, the software assumes a decimal number.

- The 64 least significant bits are programmed. The remaining most significant bits on the left are truncated.
- The software does not program decimal numbers greater than 15 digits.

Syntax option 2:

```
SIGNATURE = string
```

Examples

```
SIGNATURE = abcdefgh

SIGNATURE = ABC_123
```

Note the following.

- A string must begin with an alpha character.
- Alphanumeric characters and underscores are allowed.
- The software converts alpha characters to the corresponding ASCII code.
- Spaces are allowed in strings.
- The software converts all lower case characters that you enter into upper case characters.
- The left-most 8 characters are programmed with the corresponding ASCII code. The remaining characters on the right are truncated.

Simulation

The PRELOAD command replaces the old PRLDF command described in Chapter 4 of the PAL Data Book.

Include the PRELOAD command in the simulation segment of your PDS design file. The syntax for the PRELOAD command follows.

Syntax

```
PRELOAD    list of register identifiers
```

Example

```
PRELOAD 01 /02 03
```

The example above shows the PRELOAD command setting the register values to 101 (high, low, high).

The PRELOAD command is similar to the old PRLDF command. It forces a register into a known state, either 1 or 0. The PRELOAD statement allows you to initialize registers.

Figure 1 illustrates a PALCE16V8 output register. Notice the register is identified by the output node name A.

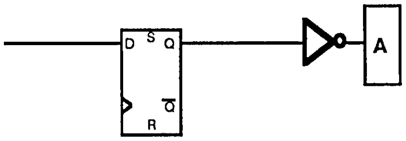


Figure 1: Output Register

To set the A register value to 1, the PRELOAD statement is written as shown in the example below.

Example

```
PRELOAD A
```

In the example above, the PRELOAD command sets the register to a value of 1. The inverter causes the output value to change to 0. Thus, the PRELOAD command determines the value of the register alone. The output value is determined by the device architecture.

Note: Unlike the PRELOAD command, the old PRLDF command determined the value of *outputs* not registers.

The example below shows a partial simulation segment for a PALCE16V8.

Example

```
CHECK O14 /O15 O16 /O17
SETF OE

PRELOAD /O14 /O15 O16 O17  ;Preload registers=0010

SETF /OE                   ;Check output
CHECK O14 O15 /O16 O17

CLOCK CLK                  ;Next state
CHECK O14 /O15 O16 /O17
```

Note: The above example assumes the pin list shown earlier in this document.

Keep the following special considerations in mind when using the PRELOAD command on the PALCE16V8.

- The register is forced to a known state and the output is calculated from the register.
- After the register is clocked, the value that represents the next state appears at the output.
- An error is generated if the output is not disabled before preload.
- The PRELOAD statement works on the register; the CHECK statement validates the output.

DESIGN A DECODER FOR THE PALCE16V8

About this Tutorial

This tutorial is a step-by-step procedure on using PALASM software to design a decoder for the PALCE16V8. It describes only those features of PALASM software that are required for the decoder design. Therefore, it works merely as an introduction to PALASM software. The full featured version of the software provides you with additional design capability for advanced applications. Contact your local Advanced Micro Devices sales office for a full-featured version of PALASM software. The software package includes a comprehensive user manual.

Prerequisites

- You need an IBM-PC/XT/AT or compatible with a hard disk.
- You need the two software disks labelled *PALCE16V8 Evaluation Kit.*
- To program the PALCE16V8 sample, a programmer must be linked up to your computer.
- To communicate with the programmer, a programmer communications program of your choice must be installed on your computer.

How to Use this Tutorial

This tutorial is designed to be read sequentially from beginning to end. First you install the software, then create a Boolean equation design, compile the design, simulate the design, and finally view the output files. The process takes approximately one hour.

Install the Software

The software on the two floppy disks labelled *PALCE16V8 Evaluation Kit* has been compressed into archive format. The installation procedure dearchives the software programs before installing them on your hard disk. This procedure takes approximately seven minutes.

Step 1. Place disk #1 in drive A.

Enter **A: INSTALL** <return>

Step 2. At the prompt, specify the drive on which you want the software installed.

Step 3. At the prompt, if necessary, allow the software to make changes to the system files AUTOEXEC.BAT and CONFIG.SYS.

Step 4. When the message window at the bottom of your screen prompts you, place disk #2 in drive A.

Step 5. When the installation procedure is complete, the following message appears on your screen.

```
Re-boot and enter C: PALC16V8
```

Follow the instructions to start up the program.

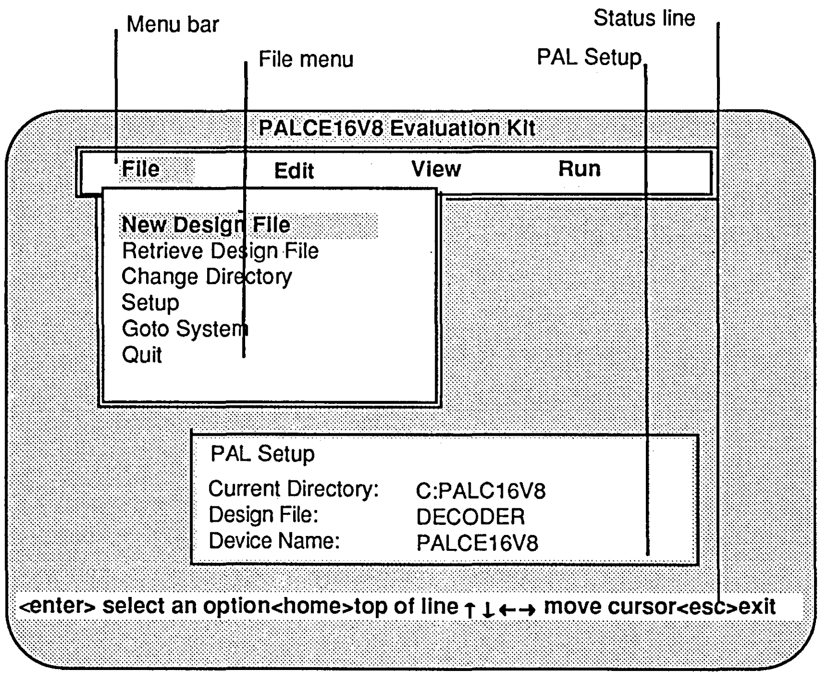
Note
The command to call up the software is PALC16V8 and not PALCE16V8.

Learn the Menu

The PALCE16V8 Evaluation Kit banner is the first screen that appears when you call up the software. When you press any key, the menu appears on your screen.

Figure 1 shows the four part software menu. These four parts, File, Edit, View and Run are arranged at the top of the screen in a menu bar.

Figure 1. PALASM Menu Screen



PAL Setup

Current Directory:	C:PALC16V8
Design File:	DECODER
Device Name:	PALCE16V8

<enter> select an option<home>top of line ↑ ↓ ← → move cursor<esc>exit

Each of the menu bar items contains a different set of program options related to that menu function. For example, the options for finding a file or a directory are located in the File menu. When you highlight one menu bar item, its menu appears. Notice that you can move laterally across the screen using the cursor movement arrows.

The status line at the bottom of the screen gives information about how to control the screen. Check this information frequently because the information changes as you perform different tasks. Use the arrow keys to move the cursor to the operation you wish to perform.

Get familiar with the menu by exploring the various options. When you are ready to begin using the software functions, proceed to *Create the Decoder Design.*

Create the Decoder Design

The simple decoder design is created in Boolean equations and implemented in a PALCE16V8 device. Your first task in creating the design is to understand the function of the decoder and interpret the function in Boolean equations.

Note
Although decoders are usually combinatorial, for the purpose of this exercise assume a registered decoder design.

The Function of the Decoder

Table 1 shows a truth table for the decoder. The decoder has three input pins: X, Y, and Z. The function of the decoder is to monitor the three input pins and assert one of eight output lines, A-H, for each of the eight combinations of inputs.

Table 1. Truth Table for Decoder

Inputs			**Outputs Generated**							
X	Y	Z	A	B	C	D	E	F	G	H
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Notice that each of the output pins is high, or has a value of 1, in response to a unique combination of the three input pins. Output pin A, for example, is high only if the three input pins are low. The Boolean expression that corresponds to this condition is

/X * /Y * /Z

Note
** is used for AND*
/ is used for NOT

You can create a Boolean equation that defines all the conditions under which output pin A is high:

A = /X * /Y * /Z

Similarly, you can create all of the Boolean equations required to completely define the decoder functions:

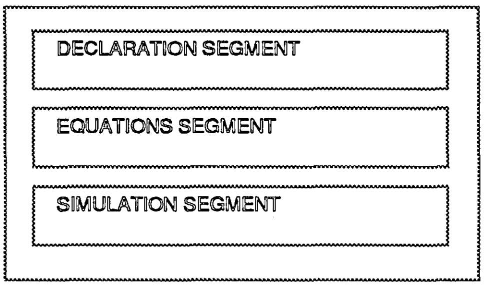
A = /X * /Y * /Z
B = /X * /Y * Z
C = /X * Y * /Z
D = /X * Y * Z
E = X * /Y * /Z
F = X * /Y * Z
G = X * Y * /Z
H = X * Y * Z

Now that you have Boolean equations to describe the decoder functions, it is time to create a complete PALASM design file for the decoder.

Learn the Structure of the PALASM Design File

PALASM software requires a specific design file layout. Figure 2 illustrates the layout. The equations defined above go into the Equations segment of the file.

Figure 2. PALASM Design File Layout



The PALASM design file is also known as the PDS (PAL device Design Specification) file.

Proceed to *Build the Declaration Segment* to begin creating the decoder design file using PALASM software.

Build the Declaration Segment

The PALASM software menu provides a template for building the Declaration segment of your design file. The procedure to use the template follows.

Step 1. Use your arrow keys to move to the File menu..

Step 2. Select New Design File in the File menu and press <return>.

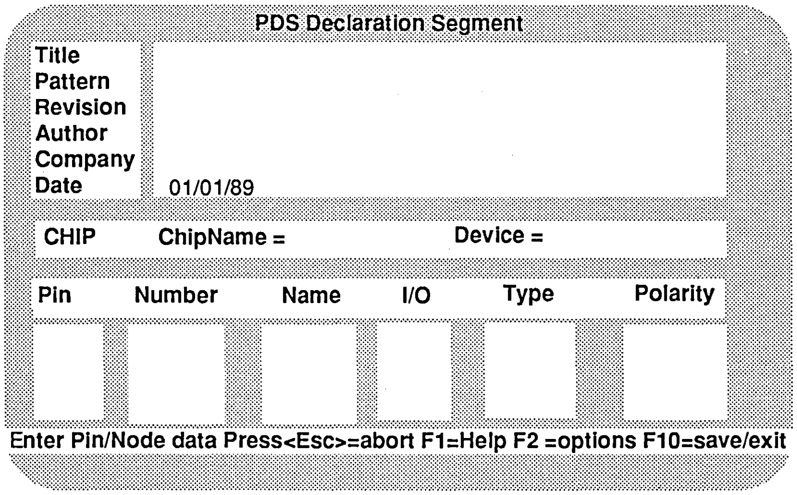
Step 3. A window appears. Enter a file name of your choice. This tutorial uses the file name shown below.

Enter **DECODER.PDS** <return>

The PDS Declaration Segment template appears on your screen.

Figure 3 shows the template as it appears on your screen. Notice you can use <return> to move from field to field. However, some fields require you to enter information before you can move on. Also, use F10 and not <return> to save the segment.

Figure 3. Screen Template of PDS Declaration Segment



PDS Declaration Segment

Title
Pattern
Revision
Author
Company
Date 01/01/89

CHIP ChipName = Device =

Pin	Number	Name	I/O	Type	Polarity

Enter Pin/Node data Press<Esc>=abort F1=Help F2 =options F10=save/exit

Enter the File Header Information

The first part of the Declaration segment consists of descriptive information about your file. You can enter the following or similar information for the decoder design. Figure 4 shows the completed file header.

Figure 4. File Header for the Decoder Design

```
TITLE        DECODER
PATTERN      A
REVISION     1.0
AUTHOR       J. ENGINEER
COMPANY      ADVANCED MICRO DEVICES
DATE         3/20/89
```

Enter the CHIP Statement

The chip name and the device name are required fields. You can enter a descriptive chip name of your choice. The software selects the device name, PALCE16V8, for you.

```
CHIP    ChipName = PAL_AMD    Device  PALCE16V8
```

Enter the Pin List

Each pin on the PALCE16V8 that you use in your design requires a pin statement. The pin statement consists of the following fields.

- Pin
- Pin number
- Pin name
- Input, output, or I/O
  Specify one of the above.
- Input or Output type
  Specify whether the input or output is combinatorial or registered.

- Polarity type
  Specify whether the output is active-low or active-high.

In *Create the Decoder Design* notice that the decoder design consists of three inputs and eight outputs. In addition, you must define pin 1 as the clock pin and pin 11 as the output enable pin.

The procedure to enter the pin list follows.

Step 1. Enter the pin statements using the arrow keys or the tab key to move from field to field. Notice that the template allows you to save time by giving you choices for several of the fields

Figure 5 shows the completed pin list.

Step 2. Press F10, *not* <return>, to save your pin list and exit the template

Figure 5. Pin List for the Decoder Design

Pin	Number	Name	I/O	Type	Polarity
Pin	1	CLOCK	Input	Comb	Active-high
Pin	2	X	Input	Comb	Active-high
Pin	3	Y	Input	Comb	Active-high
Pin	4	Z	Input	Comb	Active-high
Pin	11	OE	Input	Comb	Active-high
Pin	12	A	Output	Reg	Active-high
Pin	13	B	Output	Reg	Active-high
Pin	14	C	Output	Reg	Active-high
Pin	15	D	Output	Reg	Active-high
Pin	16	E	Output	Reg	Active-high
Pin	17	F	Output	Reg	Active-high
Pin	18	G	Output	Reg	Active-high
Pin	19	H	Output	Reg	Active-high

PALASM software transfers you to the editor and displays the file DECODER.PDS. Notice that the entire Declaration segment that you created in the template has been copied into the file. Also notice that headings of the remaining segments of the file have been entered to prompt you to complete the design file:

```
The Equations Segment

The Simulation Segment
```

Proceed to *Build the Equations Segment* to complete the next part of the decoder design file.

Build the Equations Segment

The Equations segment contains the Boolean equations that specify the decoder design.

In Table 1, the truth table defines the desired outputs A-H as a function of the inputs X, Y, and Z.

Figure 6 shows the complete equations segment for the decoder design file

Figure 6. Equations Segment for the Decoder Design

```
EQUATIONS

A = /X * /Y * /Z
B = /X * /Y *  Z
C = /X *  Y * /Z
D = /X *  Y *  Z
E =  X * /Y * /Z
F =  X * /Y *  Z
G =  X *  Y * /Z
H =  X *  Y *  Z
```

After exiting the PDS Declaration Segment template, the software displays the file DECODER.PDS on your screen. You are now in the editor. Until you quit the editor and return to PALASM, use the editor commands. The procedure to enter the Boolean equations in the Equations segment of the DECODER.PDS file follows.

Step 1. Use the arrow key to move the cursor to the line just under the keyword EQUATIONS.

Step 2. Enter the Boolean equations as shown in Figure 6. At the end of each line, press <return> to go to the next line.

Step 3. When you have entered all the equations, press <escape> to display the menu bar.

Step 4. Go to the File menu, and select Save.

Proceed to *Build the Simulation Segment* to complete the decoder design file.

Build the Simulation Segment

This segment of the design file is optional. Including the simulation segment in the design file makes simulation of the design possible. Simulation allows you to predict the behavior of your design in software. The PALASM simulator allows you to monitor the status of inputs and outputs, to control the input signals, and to check the outputs against your predicted outputs.

To simulate this design thoroughly, you must

- Set the inputs in every possible combination.
- Check if each combination of inputs produces the desired outputs.
- Supply a clock pulse to effect the change in outputs.

- Enable the outputs by setting the output enable (OE) pin low.

The simulation for the decoder design may be described in natural language as follows.

Set the output enable, clock, and input levels to /OE /CLOCK /X /Y /Z.
Supply a clock pulse.
Check that the output levels are A /B /C /D /E /F /G /H.
Set the input levels to /X /Y Z.
Supply a clock pulse.
Check that the output levels are /A B /C /D /E /F /G /H.
Set the input levels to /X Y /Z.
Supply a clock pulse.
Check if the output levels are /A /B C /D /E /F /G /H.
Set the input levels to /X Y Z.
Supply a clock pulse.
Check if the output levels are /A /B /C D /E /F /G /H.
Set the input levels to X /Y /Z.
Supply a clock pulse.
Check if the output levels are /A /B /C /D E /F /G /H.
Set the input levels to X /Y Z.
Supply a clock pulse.
Check if the output levels are /A /B /C /D /E F /G /H.
Set the input levels to X Y /Z.
Supply a clock pulse.
Check if the output levels are /A /B /C /D /E /F G /H.
Set the input levels to X Y Z.
Supply a clock pulse.
Check if the output levels are /A /B /C /D /E /F /G H.

PALASM uses simple commands to define the simulation instructions. Figure 7 shows the completed simulation segment for the decoder design in PALASM syntax.

Figure 7. Simulation Segment for the Decoder Design.

```
SIMULATION

SETF      /OE /CLOCK /X /Y /Z
CLOCKF    CLOCK
CHECK     A /B /C /D /E /F /G /H
SETF      /X /Y Z
CLOCKF    CLOCK
CHECK     /A B /C /D /E /F /G /H
SETF      /X Y /Z
CLOCKF    CLOCK
CHECK     /A /B C /D /E /F /G /H
SETF      /X Y Z
CLOCKF    CLOCK
CHECK     /A /B /C D /E /F /G /H
SETF      X /Y /Z
CLOCKF    CLOCK
CHECK     /A /B /C /D E /F /G /H
SETF      X /Y Z
CLOCKF    CLOCK
CHECK     /A /B /C /D /E F /G /H
SETF      X Y /Z
CLOCKF    CLOCK
CHECK     /A /B /C /D /E /F G /H
SETF      X Y Z
CLOCKF    CLOCK
CHECK     /A /B /C /D /E /F /G H
```

The procedure to enter the Simulation commands in the DECODER.PDS file follows.

Step 1. In the editor file, DECODER.PDS, use the arrow key to move the cursor just under the keyword SIMULATION.

Step 2. Enter the Simulation segment shown in Figure 7. At the end of each line, press <return> to go to the next line.

Step 3. When you have entered all the equations, press <escape> to display the menu bar.

Step 4. Go to the File menu, and select Save.

Step 5. Figure 8 shows you the complete decoder design file. Check your editor file to see if there are any typos. Your file will not process correctly if there are syntax errors. Correct your errors and save the file again.

Step 6. Press <esc> to call up the editor menu bar.

Step 6. Select Quit All Files in the Quit menu to return to PALASM.

The decoder design file is now complete and ready for PALASM to process. Proceed to *Process the Design File*..

Figure 8 Complete Decoder Design File

```
;PALASM Design Description

;......................Declaration Segment .......

TITLE      DECODER.PDS
PATTERN    A
REVISION   1.0
AUTHOR     J. ENGINEER
COMPANY    ADVANCED MICRO DEVICES, INC.
DATE       3/20/89

CHIP       PAL_AMD      PALCE16V8

;......................Pin Declarations ..........

PIN  1  CLK
PIN  2  X
PIN  3  Y
PIN  4  Z
PIN 11  OE
PIN 12  A  HIGH  REG
PIN 13  B  HIGH  REG
PIN 14  C  HIGH  REG
PIN 15  D  HIGH  REG
PIN 16  E  HIGH  REG
PIN 17  F  HIGH  REG
PIN 18  G  HIGH  REG
PIN 19  H  HIGH  REG
```

```
;.......................Boolean Equations Segment .

EQUATIONS

A = /X * /X * /Z
B = /X * /Y *  Z
C = /X *  Y * /Z
D = /X *  Y *  Z
E =  X * /Y * /Z
F =  X * /Y *  Z
G =  X *  Y * /Z
H =  X *  Y *  Z
```

```
;......................Simulation Segment .........

SIMULATION

SETF        /OE /CLOCK /X /Y /Z
CLOCKF      CLOCK
CHECK       A /B /C /D /E /F /G /H
SETF        /X /Y Z
CLOCKF      CLOCK
CHECK       /A B /C /D /E /F /G /H
SETF        /X Y /Z
CLOCKF      CLOCK
CHECK       /A /B C /D /E /F /G /H
SETF        /X Y Z
CLOCKF      CLOCK
CHECK       /A /B /C D /E /F /G /H
SETF        X /Y /Z
CLOCKF      CLOCK
CHECK       /A /B /C /D E /F /G /H
SETF        X /Y Z
CLOCKF      CLOCK
CHECK       /A /B /C /D /E F /G /H
SETF        X Y /Z
CLOCKF      CLOCK
CHECK       /A /B /C /D /E /F G /H
SETF        X Y Z
CLOCKF      CLOCK
CHECK       /A /B /C /D /E /F /G H
```

Process the Design File

Before you begin processing the decoder design created in the previous sections, take a look at the software processing sequence.

The processing sequence consists of two simple steps.

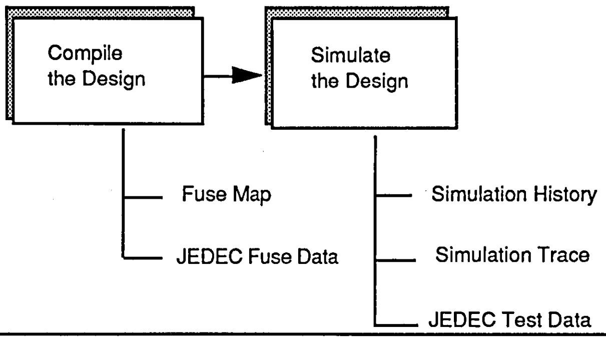
1. Compile the design and generate JEDEC output.

2. Simulate the design.

The main purpose of using PALASM is to translate your input design into programmer-readable JEDEC output. However, through simulation, PALASM allows you to test your design without actually programming a device.

Figure 8 illustrates the software processing sequence. Notice that both the compile and Simulation processes generate output files.

Figure 9. PALASM Software Processing Sequence

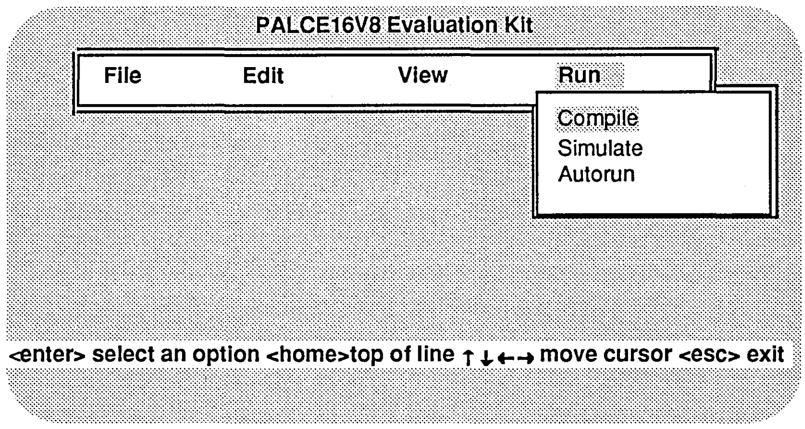


Now that you have an overview of the procedure, you can begin processing the decoder design. Proceed to *Autorun Compile and Simulate.*

Autorun Compile and Simulate

Use the arrow keys to go to the Run menu in PALASM. Notice that the Run menu offers you three choices. Figure 9 illustrates the Run menu.

Figure 10. Run Menu


PALCE16V8 Evaluation Kit
File
Edit
View
Run
Compile
Simulate
Autorun
<enter> select an option <home>top of line ↑↓←→ move cursor <esc> exit

PALASM software offers you a time saving autorun feature that combines the compile and Simulation processes into one keystroke. The autorun procedure follows.

Step 1. Select Autorun in the Run menu
A window opens at the bottom of your screen.

Step 2. Watch the status line as PALASM software completes the following operations.

Parse

Minimize

Assemble

Simulate

Step 3. When you see the message

```
PLDSIM Program Successful
```

Press <esc>.

If the process was successful, you can skip steps 4-11.

If the process was unsuccessful and produces errors, proceed to step 4.

Step 4. Select Edit in the PALASM menu bar. The Edit menu appears on your screen.

Step 5. Select Design File in the Edit menu. The design file DECODER.PDS appears.

Step 6. Carefully compare the file on your screen with the printed file in Figure 8, *Complete Decoder Design File.* If your have typos in your screen file, make the necessary changes.

Step 7. Press <esc> to display the editor menu bar.

Step 8. Select File in the menu bar. The File menu appears.

Step 9. Select Save in the File menu.

Step 10. To quit the editor, select Quit in the menu bar. The Quit menu appears.

Step 11. Select Quit All Files. The software returns you to the PALASM menu.

Step 12. Now repeat steps 1-3 to recompile and simulate your design file.

Note
The decoder design has been tested and found error-free. If your compile and simulation processes produce errors, you probably have typos in your file.

Now that the design file has been successfully processed, you can look at the output files that the compile and simulation processes generated. Proceed to *View the Output Files.*

View the Output Files

In the last section you used autorun to run the compile and Simulation processes with one keystroke. PALASM, however, generates a set of output files after each process. Proceed to view each set of output files.

The procedure to view any of the output files follows.

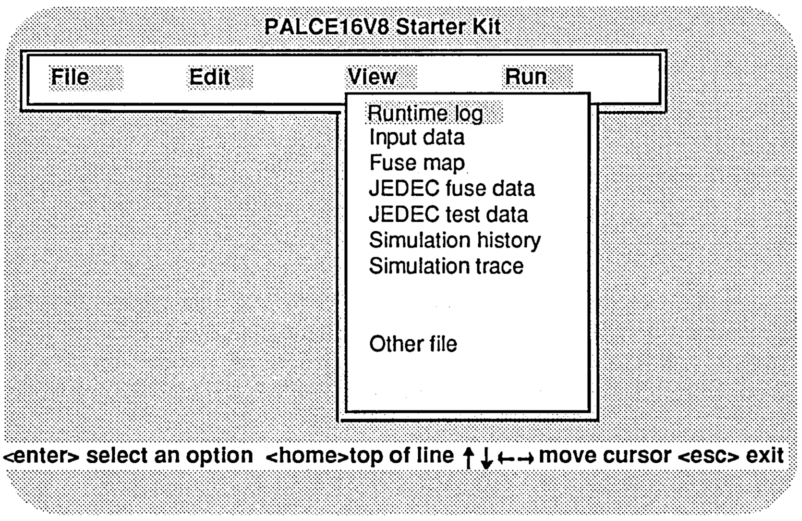
Step 1. Use the arrow keys to select the View menu. Figure 11 shows the View menu as it appears on your screen

Step 2. Notice that the list contains input, output and intermediate files. To view a file, select the item and press <return>

Step 3. The file is now displayed on your screen. Notice you can scroll up and down using the arrow keys.

Step 4. Press <esc> to exit the file.

Figure 11. View Menu



View the Compile Output Files

The compile process generates the following output files.

• The Fuse Map	*Decoder.XPT*
• The JEDEC Fuse Data	*Decoder.JED*

Notice the file names, shown above in italic, that PALASM assigns the output files. The first part of the name is user-defined. The second part is the extension that the software assigns.

The Fuse Map

The fuse map is a detailed map of the connections that are programmed along each product term on the device. The following symbols illustrate which connections are programmed and unprogrammed.

x	Unprogrammed connection
–	Programmed connection

The JEDEC Fuse Data

This file is the programmer-readable translation of the input design file. It can be downloaded to the programmer to program the PALCE16V8. JEDEC stands for Joint Electronic Device Engineering Council, the organization that creates the standards for this file.

View the Simulation Output Files

The Simulation output files show you whether your design produces the desired outputs. The Simulation process produces the following output files.

• Simulation History	*Decoder.HST*
• Simulation Trace	*Decoder.TRF*
• JEDEC Test Data	*Decoder.JDC*

The decoder design does not use the TRACE command. Therefore, the trace file is not generated and is not discussed in this tutorial.

The Simulation History

The simulation history shows the status of all the signals defined in the pin list. It uses symbols to represent the different states:

H	High
L	Low
X	Undefined
Z	Output disabled

Figure 12 shows a sample history file.

Figure 12. Sample Simulation History File

```
        "g" represents                 "c" represents
           SETF                           CLOCKF

       g cg cg cg cg cg cg cg c
CLOCK  LHLLHLLHLLHLLHLLHLLHLLHL
X      LLLLLLLLLLLLHHHHHHHHHHHH
Y      LLLLLLHHHHHHLLLLLLHHHHHH
Z      LLLHHHLLLHHHLLLHHHLLLHHH
OE     LLLLLLLLLLLLLLLLLLLLLLLL
A      HHHLLLLLLLLLLLLLLLLLLLLL
B      LLLHHHLLLLLLLLLLLLLLLLLL
C      LLLLLLHHHLLLLLLLLLLLLLLL
D      LLLLLLLLLHHHLLLLLLLLLLLL
E      LLLLLLLLLLLLHHHLLLLLLLLL
F      LLLLLLLLLLLLLLLHHHLLLLLL
G      LLLLLLLLLLLLLLLLLLHHHLLL
H      LLLLLLLLLLLLLLLLLLLLLHHH
```

The JEDEC Test Data

The simulation process generates test vectors that are added to the JEDEC file discussed in *The JEDEC File* above. The test vectors can be used to test and verify the design on the device programmer.

This completes the PALASM design and simulation process. The next step is to download your JEDEC file to a device programmer. Consult the Programmers Development Systems Table in this data sheet part of this document for information on programmers. Also, refer to your programmer manual for instructions on setup and use.

Where to Go from Here

This tutorial did not explore all the capabilities of PALASM software or the PALCE16V8. To order the full-featured version of PALASM software, contact your local AMD sales office today.