Simple Systems Interface for UltraNAND™ Flash

Application Note

July 2003

The following document refers to Spansion memory products that are now offered by both Advanced Micro Devices and Fujitsu. Although the document is marked with the name of the company that originally developed the specification, these products will be offered to customers of both AMD and Fujitsu.

Continuity of Specifications

There is no change to this document as a result of offering the device as a Spansion product. Any changes that have been made are the result of normal documentation improvements and are noted in the document revision summary, where supported. Future routine revisions will occur when appropriate, and changes will be noted in a revision summary.

Continuity of Ordering Part Numbers

AMD and Fujitsu continue to support existing part numbers beginning with "Am" and "MBM". To order these products, please use only the Ordering Part Numbers listed in this document.

For More Information

Please contact your local AMD or Fujitsu sales office for additional information about Spansion memory solutions.

Publication Number **22363** Revision **A** Amendment **0** Issue Date **April 20, 1999**

Simple System Interface for UltraNAND™ Flash

Application Note

AMD has developed the UltraNAND™ product line to address high-density non-volatile memory needs. Target applications include code and data storage in embedded or removable media systems. This application note describes a simple hardware system interface for up to three UltraNAND banks, using a single AmPALLV16V8-10SC or AmPALLV22V10-10PC PLD (Programmable Logic Device). A bank combines two or more 8-bit UltraNAND devices to accomodate system system bus width requirements (16-bit or higher).

Note: *Please refer to the UltraNAND device data sheet and "Boot Loader for UltraNAND Flash Simple System Interface" application note for further information as necessary.*

Appropriate Applications

AMD's UltraNAND Flash provides high-speed read, program, and erase operations at a lower cost per bit than NOR Flash, which makes it ideal for high-density non-volatile storage applications. Appropriate applications are those that use non-volatile memory to store code or data which is transferred, or shadowed, to a high-speed memory resource—like synchronous DRAM—for fast random access or execution. UltraNAND has relatively slow (7 µs) access to random pages, but fast (50 ns) access to sequential bytes within a selected page. The slow random access and the requirement for command and address input makes UltraNAND inappropriate for code XIP (Execute in Place) operations. The fast sequential read and write speed, high-density, and lower cost per bit make UltraNAND ideal for disk replacement or other high-volume non-volatile storage applications that do not require high-speed random access.

System Related Benefits of UltraNAND

UltraNAND has been designed to be fully hardware and software compatible with NAND architecture Flash already available on the market. However, current competitive products generally require ECC (Error Check and Correction) to meet their specified program/erase cycle endurance, and are not typically available as 100% good devices. Most of these competing devices are sold with the understanding that they may have a few bad blocks when shipped from the manufacturer. Using a product with bad blocks usually requires that the system provide some form of bad block management to map defective blocks out of the system memory space. The requirement for ECC and bad block mapping adds hardware and software design complexity, generally increases total system cost, and impacts system performance. In some cases the performance impact can be as great as a 40% decrease in overall data throughput.

UltraNAND has been designed as an improved product solution over the competition, providing 100,000 program/erase cycle endurance without requiring ECC. UltraNAND is also available with 100% good blocks, which eliminates the need for bad block mapping. The program/erase cycle endurance, without ECC, and availability of 100% good devices greatly simplifies system design requirements. Applications can use UltraNAND with simple system interface logic, rather than the sophisticated memory controller that would be required with competing devices. The package, pinout, command set, and bus interface compatibility of UltraNAND allow applications designed to support older NAND architecture devices to be simplified, and/or cost reduced, with UltraNAND.

UltraNAND Interface Requirements

As shown in Figure 1, UltraNAND utilizes a multiplexed command/address/data bus. All command, address, and data information is passed to and from the device through I/O[0..7] (8-bit I/O port). Control signals are provided on the device for CE# (Chip Enable), CLE (Command Latch Enable), ALE (Address Latch Enable), WE# (Write Enable), RE# (Read Enable), SE# (Spare area Enable), and WP# (Write Protect). There is also an open drain RY/BY# (Ready/Busy) output pin used to indicate when the device is busy with an internal operation.

System applications using UltraNAND must generate the proper control signals for the device which, in many cases, are not used by any other system resource. Fortunately, the interface logic required to generate the appropriate UltraNAND signals is reasonably simple and can be designed in discrete logic, incorporated in a PLD, or included in the system ASIC. The simple system interface designs included in this application note have been developed for the AmPALLV16V8 and AmPALLV22V10 PLDs. Source and simulation code for two PLD versions, single and multiple bank support,

Publication# **22363** Rev: **A** Amendment/**0**
Issue Date: **April 20, 1999**

is included in this application note. The PLDs generated are shown in Figure 4 and Figure 5.

System Interface Description

Using Programmable I/O

In many applications using UltraNAND, there may be programmable I/O lines available, either in a microcontroller or the system interface logic, that can be used to generate the UltraNAND CE#, CLE, ALE, SE#, and WP# inputs. With enough programmable I/O lines available, the Interface PLD described in this application note is not required, and the system can provide all of the interface support needed through those programmable I/O lines. If WP# does not need to be dynamically controlled by the system, only four programmable I/O lines are required. This application note is intended to describe the basic logic needed to control up to three UltraNAND banks for those applications that do not have programmable I/O lines available.

Typical System Implementation

A typical application of the simple system interface is shown in Figure 1 with the Interface PLD supporting a single UltraNAND bank consisting of two UltraNAND devices. The Interface PLD uses the usual system control signals to create the full set of control signals needed by the UltraNAND Flash. The system software is then able to program the Interface PLD to generate the signal sequence required for proper operation. Because fewer than three UltraNAND banks are required, the 16V8 PLD version that supports only one bank was used.

The INIT signal is unused in this application example. Refer to the “Boot Loader for UltraNAND Flash Simple System Interface” application note for information on generating an INIT signal using a Boot Loader PLD.

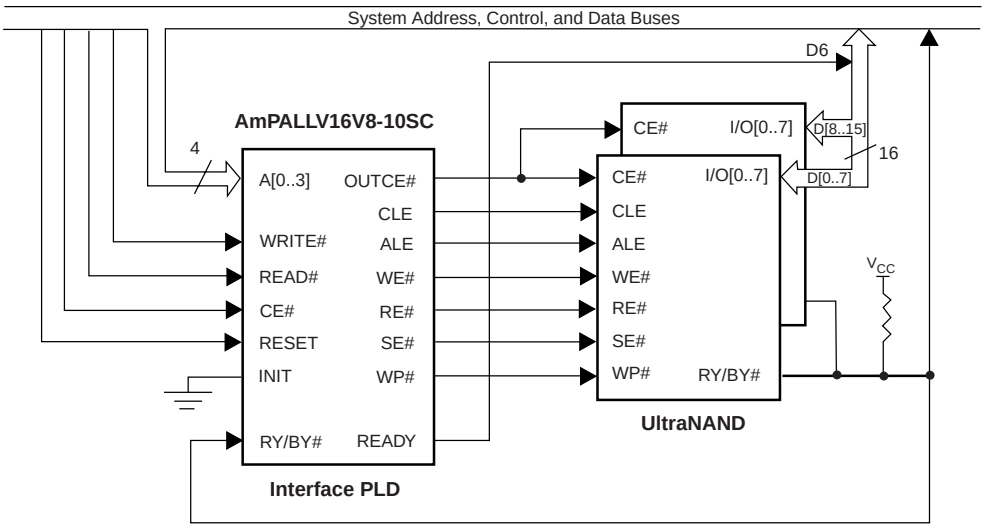


Figure 1. Typical System Interface Application for a Single UltraNAND Bank

System Interface Signal Description

The UltraNAND Interface PLD is responsible for generating the signals required by the UltraNAND device, which are not found in typical systems. These are CLE, ALE, WE#, RE#, SE#, and WP#. In order to control the Interface PLD, the system is responsible for providing some of the more typical address and control signals. The definition of all pertinent signals, and the source required to generate the signals, are included in Table 1. A timing diagram of a Read First Half Page operation is shown in Figure 2, and includes all of the signals required by UltraNAND.

Table 1. UltraNAND System Interface Signal Description

Signal	Source	Definition
INIT	Optional Boot Loader PLD	From optional “Boot Loader for Simple System Interface” PLD to indicate that the Boot Loader PLD is initializing UltraNAND
ALE	Interface PLD	Address Latch Enable required for address latch cycles
CLE	Interface PLD	Command Latch Enable required to latch the command
OUTCE[0..2]#	Interface PLD	The Chip Enables used to select the UltraNAND bank(s)
RE#	Interface PLD	Read signal used by UltraNAND for all read cycles
READY	Interface PLD	Tri-state output to allow the system to read the state of the RY/BY# pin(s)
SE#	Interface PLD	Spare Area Enable to control access to the UltraNAND spare area
WE#	Interface PLD	Write signal used by UltraNAND for all write cycles
WP#	Interface PLD	Write Protect to prevent program/erase operations in UltraNAND
A[0..3]	System	Address bus used to select one of sixteen Interface PLD control ports
CE#	System	The Chip Enable used to select the UltraNAND and Interface PLD
I/O[0..7]	System/UltraNAND	The eight I/O lines used to transfer commands, addresses, and data
RESET	System	A system reset signal which remains high until V_{CC} is valid
READ#	System	Read signal used by the system for all read cycles
WRITE#	System	Write signal used by the system for all write cycles
RY/BY#	UltraNAND	Ready/Busy signal to indicate the current state of UltraNAND

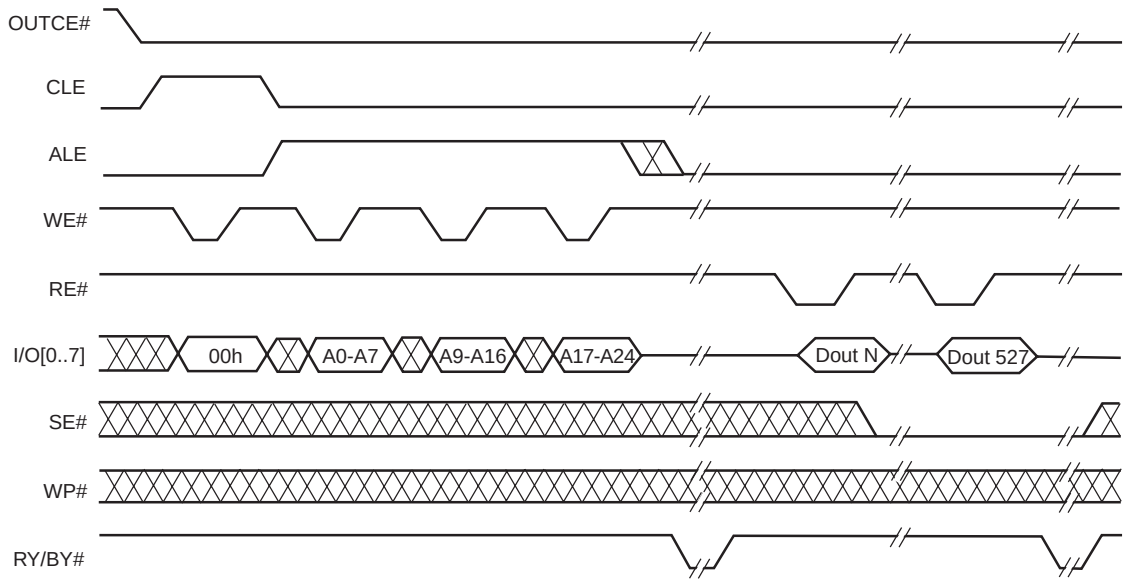


Figure 2. UltraNAND Read First Half Page Timing Diagram

Interface PLD Port Addresses

The UltraNAND Interface PLD occupies sixteen port addresses. The port address locations are determined by the CE# (Chip Enable) signal generated by the system. This allows the UltraNAND device, or devices, to be set up as either a memory mapped or I/O mapped system resource. The system sets the base address of the device through the chip enable address decode and the UltraNAND interface will then be accessible at that address, and through the next fifteen sequential addresses. As an example, if the chip enable decode is set for I/O location 200h, the Interface PLD would reside at I/O locations 200h through 20Fh. A description of each interface address and the operation performed is included in Table 2.

Table 2. UltraNAND System Interface Control Port Definition

Port	Read	Write	Operation Performed
0	Data, ID, or Status	Address or Data	Read information depends on previous command loaded Write addresses (ALE = high) or data (ALE = low)
1	N/A	Command	All commands are written through this port with ALE cleared (low)
2	N/A	Set ALE	Set ALE (high) to allow addresses to be written
3	N/A	Clear ALE	Clear ALE (low) to allow commands, or data, to be written
4	N/A	Set SE#	Set SE# (low) to allow access to the spare area of each page
5	N/A	Clear SE#	Clear SE# (high) to prevent access to the spare area of each page
6	N/A	Set WP#	Set WP# (low) to prevent program/erase cycles
7	N/A	Clear WP#	Clear WP# (high) to allow program/erase cycles
8	N/A	Set CE0#	Set CE0# (low) to enable the first UltraNAND bank
9	N/A	Clear CE0#	Clear CE0# (high) to disable the first UltraNAND bank
A	N/A	Set CE1#	Set CE1# (low) to enable the second UltraNAND bank
B	N/A	Clear CE1#	Clear CE1# (high) to disable the second UltraNAND bank
C	N/A	Set CE2#	Set CE2# (low) to enable the third UltraNAND bank
D	N/A	Clear CE2#	Clear CE2# (high) to disable the third UltraNAND bank
E	N/A	N/A	No Function
F	RY/BY# Status	N/A	Read the state of all RY/BY# pins through this port

Legend: *X = Don't Care*

Interface PLD Theory of Operation

This section describes the functional blocks of the Interface PLD. The example AmPALLV16V8 and AmPALLV22V10 PLDs generated are shown in Figure 4 and Figure 5 respectively. An application schematic for three UltraNAND devices, supported by a single AmPALLV22V10 PLD, is shown in Figure 6.

Port Decode

In the simple system interface design PLD examples, combinatorial logic is used to perform the basic port address decode function. This allows the Interface PLD to respond to address decodes 00h through 0Fh as an offset from the base address determined by the CE# input. The decoder logic outputs are used to determine when to drive the CLE, RE#, and WE# signals to UltraNAND directly, to set and clear transparent latches for ALE, OUTCE[0..2]#, SE#, and WP#, or to allow the system to read the state of the RY/BY# pin through the Interface PLD READY output.

Output Signal Generation

The simple system Interface PLDs use combinatorial logic and transparent latches to generate all of the signals required by UltraNAND. CLE, RE#, and WE# are generated dynamically and are not latched. ALE, OUTCE[0..2]#, SE#, and WP# are latched in the design so that the system has the ability to set or clear these signals as needed.

- CLE is a dynamic signal which is generated whenever the system performs an access to port 01h. This causes CLE to go active during both reads and writes to port 01h. When the system writes a command out to port 01h, the command on the UltraNAND I/O bus will be written with CLE active, allowing the command to be accepted by the device. Prior to a command being written to the device ALE must be cleared (ALE = low) and the chip enable for the appropriate UltraNAND device must be latched low.
- ALE is a latched signal which may be set or cleared under software control and determines whether addresses, commands, or data are written to UltraNAND. ALE must be cleared when a command or data is to be written. Once ALE is set, the address may be written to port 00h to load the address into UltraNAND. ALE must be set prior to the first address cycle written and must be cleared following the last address write cycle. In the Interface PLD, the latch is set and cleared by writing to ports 02h and 03h respectively. Only a write to the port is required and the data written is irrelevant. In the PLDs, ALE is latched low (cleared) following power-up.
- OUTCE[0..2]# are latched signals which may be set or cleared under software control and determine whether an UltraNAND device is enabled or disabled. The appropriate chip enable output must be set when a command, address, or data is to be written to any UltraNAND device. The appropriate chip enable output must also be set for any read operation, and must be held asserted during all read latency periods to avoid aborting the requested read operation. In the Interface PLD, the latch for OUTCE0# is set and cleared by writing to ports 08h and 09h, the latch for OUTCE1# is set and cleared by writing to ports 0Ah and 0Bh, and OUTCE2# is set and cleared by writing to ports 0Ch and 0Dh respectively. Only a write to the port is required and the data written is irrelevant. In the PLDs, all OUTCE[0..2]# outputs are high (cleared) following power-up.
- RE# is a dynamic signal which is generated whenever the system performs a read from the Interface PLD port 00h or 0Fh. This allows the system to read Data, ID, or Status information from UltraNAND through port 00h, or to read the state of the RY/BY# pin(s) through the Interface PLD port 0Fh.
- READY is a dynamic, tri-state output from the Interface PLD that allows the system to read the state of the UltraNAND RY/BY# pin(s). The Interface PLD will drive the state of the RY/BY# pin(s) to the READY pin when a read from Interface PLD port 0Fh is performed.
- SE# is a latched signal which may be set or cleared under software control and determines if the spare area of each page (bytes 512 – 527) can be read or written. Once SE# is set, the system may read or write to the spare area through port 00h. SE# must be set prior to the Read Spare Area command (50h) being written to the device. If the spare area is to be read or written along with the normal Flash page information (bytes 0 – 511) then SE# must be set at least two cycles before the first spare area address (512) is accessed. In the PLD, SE# is set and cleared by writing to ports 04h and 05h respectively. Only a write to the port is required and the data written is irrelevant. SE# is latched low (set) following power-up.
- WE# is a dynamic signal which is generated whenever the system performs a write to the Interface PLD port 00h or 01h. This allows the system to write Address or Data information to UltraNAND through port 00h, or to write commands to UltraNAND through the Interface PLD port 01h.
- WP# is a latched signal which is set or cleared under software control, and determines whether or not program or erase operations are allowed in UltraNAND. Once WP# is set (low) the system is unable to program or erase any location in the device, and only read cycles are allowed. In the PLD, the WP# latch is set or cleared by a write to port 06h or 07h respectively. Only a write to the port is required and the data written is irrelevant.

  WP# is latched low (set) following power-up. A RESET signal, which remains low until power is valid, should be used to guarantee that WP# is asserted during power transitions. This is a requirement of UltraNAND and NAND technology devices, and is used to guarantee that there is no data corruption in the device during power transitions.

Other Signals (Generated by the System Logic)

- A[0..3] are the four least significant address bits, generated by the system, which are used to select one of sixteen ports in the Interface PLD.
- CE# is generated by the system logic and determines the address that the Interface PLD and UltraNAND devices will occupy. The system may assign the CE# decode address to be any consecutive sixteen port address in either the I/O or memory address space.
- INIT is generated by an optional "Boot Loader" PLD which is only required if the system needs to read from the device immediately after power-up. The boot loader issues a Gapless Read (02h) command sequence to the UltraNAND device, or devices, to pre-load the data registers of each device with the first page of information.
- READ# is the system generated read strobe that is used to read from the Interface PLD or UltraNAND devices. In addition to the CE# input, the READ# signal determines if the Interface PLD and UltraNAND read operations are I/O or memory mapped.
- RESET is generated by the system and is used to initialize the Interface PLD. The RESET signal should be asserted (high) whenever the system power is ramping up or down. This allows the Interface PLD to assert WP# to protect the UltraNAND devices from data corruption during power transitions.
- RY/BY# is generated by the UltraNAND device to indicate when the device is busy with an internal operation. The system may use the RY/BY# hardware signal, or poll the RY/BY# status bit in the status register, to determine when an operation is in process, or has completed.
- WRITE# is the system generated write strobe used to write to the Interface PLD or UltraNAND devices. In addition to the CE# input, the WRITE# signal is used to determine if the Interface PLD and UltraNAND write operations are I/O or memory mapped.

Design Notes

With the simple interface described in this application note, the system has full control of up to three UltraNAND devices. There are a number of basic modifications that can be made to the design to allow different system configurations to be supported.

Modifications to WP# Circuit

In many applications, the WP# input to UltraNAND is not dynamically controlled by the system. Instead it is controlled by a jumper, or switch, that allows the circuit to be hardware protected against program or erase cycles. For this modification to the simple system interface, the PLD output for WP# would be replaced with a jumper or switch. Again, some form of RESET signal must be used to force WP# active (low) during power transitions.

Removing WP# from the Interface PLD would free up an output pin and would allow a fourth chip enable output to be supported. This would allow the Interface PLD to support up to four banks of UltraNAND instead of only three banks.

Supporting Multiple UltraNAND Banks

The simple system interface provided supports up to three UltraNAND banks with all of the signals generated by the Interface PLD, except the output chip enables common to all devices. The diagram in Figure 3 shows how two banks, with two UltraNAND devices in each bank, can be supported. In this application, the WP# signal is common to all of the Flash devices so that the entire array is either protected or unprotected as a block. If preferred, the logic to generate WP# could be modified to provide a separate WP# for each device.

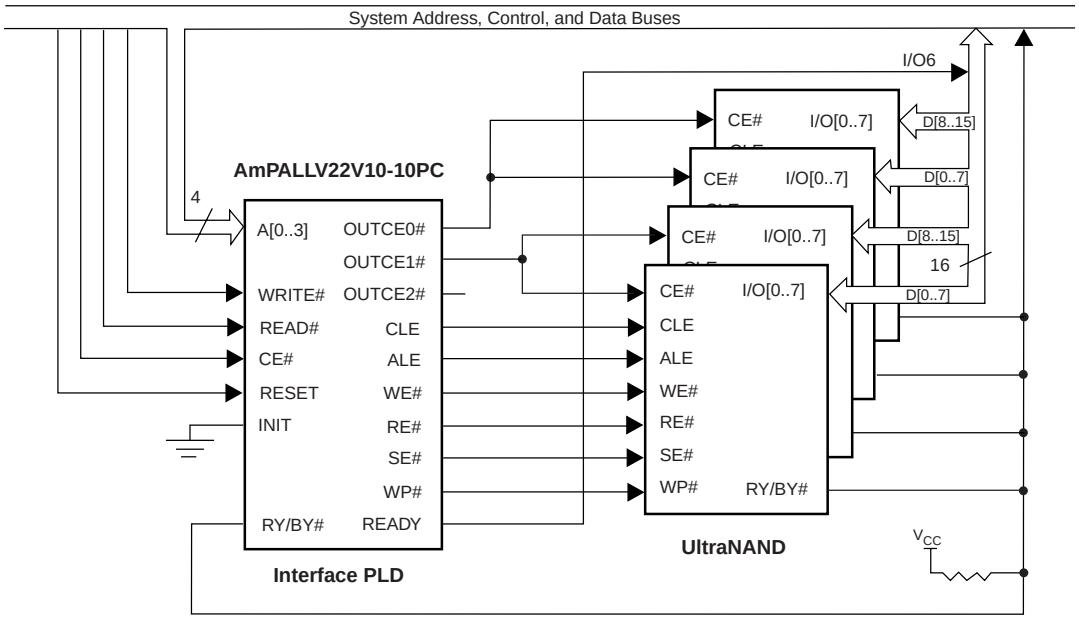


Figure 3. Supporting Multiple UltraNAND Banks

Pseudo Code Example

Included below is an example of how the Interface PLD can be used to allow the system software to control all operations that UltraNAND supports. In this example, one 528 byte page of information will be input to one device and then programmed. The following system configuration is assumed:

1. SOURCE – This is a 528 byte source buffer in system memory
2. DEST – This is the target page to be programmed in UltraNAND
3. PORTADDR – This is a system generated I/O mapped port address for the UltraNAND device to occupy

```
INIT:
    SOURCE ADDRESS POINTER                  ;Set up the address for the 528 byte source system page
    DEST ADDRESS POINTER                    ;Set up the address for the 528 byte target page in Flash
START:
    Write 00h to (PORTADDR + 8)             ;Set OUTCE0# (low) to enable UltraNAND.  Data is a don't care
    Write 00h to (PORTADDR + 3)             ;Clear ALE (low) prior to issuing command.  Data is a don't care
    Write FFh to (PORTADDR + 1)             ;Send a reset command to reset the Flash (may be omitted)
    Write 00h to (PORTADDR + 1)             ;Send a "Read First Half Page" command to the Flash to set
                                            ;the internal pointer to the first half page region in the Flash
    Write 00h to (PORTADDR + 4)             ;Set SE# (low) to allow Spare Area access.  Data is a don't care
    Write 00h to (PORTADDR + 7)             ;Clear WP# (high) to allow Flash program.  Data is a don't care
    Write 80h to (PORTADDR + 1)             ;Send an "Input Data" command to the Flash
    Write 00h to (PORTADDR + 2)             ;Set ALE (high) prior to issuing addresses.  Data is a don't care
    Write DEST[A7-A0] to (PORTADDR + 0)     ;Load the first address byte into the Flash
    Write DEST[A16-A9] to (PORTADDR + 0)    ;Load the second address byte into the Flash
    Write DEST[A24-A17] to (PORTADDR + 0)   ;Load the third address byte into the Flash
    Write 00h to (PORTADDR + 3)             ;Clear ALE (low) prior to writing data.  Data is a don't care
LOOP1:                                      ;This is where we fill the Flash buffer
    Write [SOURCE] to (PORTADDR + 0)        ;Write the data contents at the SOURCE location to the data port
    Does SOURCE = 527                       ;Has the last data byte been written?
        IF YES branch to PROG:              ;If the last byte is loaded go ahead and program the Flash
    SOURCE = SOURCE + 1                     ;If not point to the next address location and write another byte
    Branch to LOOP1:
PROG:                                       ;All 528 bytes are loaded so program the Flash
    Write 10h to (PORTADDR + 1)             ;Send a "Page Program" command to the Flash
    Write 70h to (PORTADDR + 1)             ;Send a "Read Status" command to the Flash
CHKSTAT:
    Read from (PORTADDR + 0)                ;Read the device status to see if the program is done
    Does D6 = 1                             ;Check to see if the device is ready
        IF YES branch to DONE:              ;If the program operation is done go and check for pass/fail
    Branch to CHKSTAT:                      ;If not continue to check the status for a ready condition
DONE:
    Write 00h to (PORTADDR + 6)             ;Set WP# (low) to re-protect the Flash.  Data is a don't care
    Read from (PORTADDR + 0)                ;Read the device status again to see if the program passed
    Does D0 = 1                             ;Check to see if the program operation passed or failed
        IF YES branch to FAIL:              ;A "1" in the D0 location indicates a failure condition
    Write 00h to (PORTADDR + 9)             ;Clear OUTCE0# (high) to disable UltraNAND.  Data is a don't care
    Return (program successful)             ;The program operation passed so return and report
FAIL:
    Write 00h to (PORTADDR + 9)             ;Clear OUTCE0# (high) to disable UltraNAND.  Data is a don't care
    Return (program failed)                 ;The program operation failed so return and report
```

PLD Example Implementation

PLD Source Code for UNISA16

This Interface PLD, designed into a single AmPALLV16V8, allows one UltraNAND bank to be supported by a single interface chip. This example includes the PLD source code and the simulation file used to verify the design. The AMD AmPALLV16V8-10SC used in the example powers up in a known state which covers initial power-up conditions. The RESET input is used to initialize the transparent latches in the PLD. Figure 4 is a diagram showing the pin-out of the AmPALLV16V8-10SC used in this design example.

```
Name         UNISA16;
Partno       PLD001;
Date         03/23/99;
Revision     0;
Designer     Ralph Gibson;
Company      Advanced Micro Devices;
Assembly     UltraNAND(TM) ISA Development Board Interface Solution;
Location     U1;
Device       G16V8;
Format       J;

/*************************************************************************\
**  This is a simple interface for UltraNAND allowing one UltraNAND to be   **
**  supported on a PC ISA bus. This device includes a RESET input to force  **
**  WP# asserted on power transitions.  RESET is high until Vcc is valid    **
**  and goes high when supply power ramps down.                             **
\*************************************************************************/

/**  Inputs  **/

Pin 1        = !WRITE;                              /* System Write Enable */
Pin 2        = !READ;                               /* System Read Enable */
Pin [3..6]   = [A0..3];                    /* Address input to select port */
Pin 7        = !CE;          /* Chip Enable for the interface and UltraNAND */
Pin 8        = RY_BY;                        /* RY/BY# input from UltraNAND */
Pin 9        = RESET;       /* RESET - high for reset and power transitions */
Pin 11       = INIT;              /* INIT from the optional Boot Loader PLD */

/**  Outputs  **/

Pin 19       = READY;             /* Allows system to read RY/BY# pin state */
Pin 18       = CLE;                    /* Command Latch Enable to UltraNAND */
Pin 17       = ALE;                    /* Address Latch Enable to UltraNAND */
Pin 16       = !SE;                       /* Spare Area Enable to UltraNAND */
Pin 15       = !WP;                           /* Write Protect to UltraNAND */
Pin 14       = !OUTCE;                          /* Chip Enable to UltraNAND */
Pin 13       = !WE;                            /* Write Enable to UltraNAND */
Pin 12       = !RE;                             /* Read Enable to UltraNAND */
```

```
/** Declarations and Intermediate Variable Definitions **/

PORT0 =    CE & !A3 & !A2 & !A1 & !A0              /* Data read/write port */
      #  INIT & !A1 & !A0;
PORT1 =    CE & !A3 & !A2 & !A1 &  A0                   /* CLE write port */
      #  INIT & !A1 &  A0;
PORT2 =    CE & !A3 & !A2 &  A1 & !A0                  /* Used to set ALE */
      #  INIT &  A1 & !A0;
PORT3 =    CE & !A3 & !A2 &  A1 &  A0                 /* Used to clear ALE */
      #  INIT &  A1 &  A0;
PORT4 =    CE & !A3 &  A2 & !A1 & !A0;                /* Used to set SE# */
PORT5 =    CE & !A3 &  A2 & !A1 &  A0;               /* Used to clear SE# */
PORT6 =    CE & !A3 &  A2 &  A1 & !A0;                /* Used to set WP# */
PORT7 =    CE & !A3 &  A2 &  A1 &  A0;               /* Used to clear WP# */
PORT8 =    CE &  A3 & !A2 & !A1 & !A0;              /* Used to set OUTCE# */
PORT9 =    CE &  A3 & !A2 & !A1 &  A0;             /* Used to clear OUTCE# */
PORTA =    CE &  A3 & !A2 &  A1 & !A0;                      /* No Function */
PORTB =    CE &  A3 & !A2 &  A1 &  A0;                      /* No Function */
PORTC =    CE &  A3 &  A2 & !A1 & !A0;                      /* No Function */
PORTD =    CE &  A3 &  A2 & !A1 &  A0;                      /* No Function */
PORTE =    CE &  A3 &  A2 &  A1 & !A0;                      /* No Function */
PORTF =    CE &  A3 &  A2 &  A1 &  A0;             /* To read RY/BY# state */

/**  Logic Equations  **/

READY.OE = PORTF & READ; /* READY is only driven during a PORTF read */

READY    = RY_BY; /* READY shows the state of RY/BY# */

CLE      = PORT1; /* Assert CLE on all PORT1 accesses */

ALE      = !RESET & ( WRITE & PORT2          /* Latch ALE on write to PORT2 */
                  # ALE & !( WRITE & PORT3 ) /* and clear on write to PORT3 */
                  # ALE & PORT2 );          /* Transparent latch cover term */

SE       = RESET                             /* Latch SE# asserted on RESET */
         # WRITE & PORT4                     /* Latch SE# on write to PORT4 */
         # SE & !( WRITE & PORT5 )           /* and clear on write to PORT5 */
         # SE & PORT4;                      /* Transparent latch cover term */

WP       = RESET                             /* Latch WP# asserted on RESET */
         # WRITE & PORT6                     /* Latch WP# on write to PORT6 */
         # WP & !( WRITE & PORT7 )           /* and clear on write to PORT7 */
         # WP & PORT6;                      /* Transparent latch cover term */

OUTCE    = !RESET & ( INIT                 /* Latch OUTCE# asserted on INIT */
                  # WRITE & PORT8     /* Latch OUTCE# on write to PORT2 and */
                  # OUTCE & !( WRITE & PORT9 )   /* clear on write to PORT3 */
                  # OUTCE & PORT8 );        /* Transparent latch cover term */

WE       = WRITE & ( PORT0     /* Drive WE# to UltraNAND for PORT0 or PORT1 */
                 # PORT1 );

RE       = READ & PORT0;           /* Drive RE# to UltraNAND for PORT0 only */
```

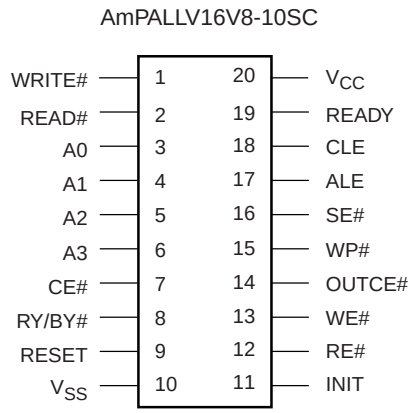
PLD Simulation File for UNISA16

The simulation file included here is used to verify the design of the PLD supporting one UltraNAND bank.

```
Name       UNISA16;
Partno     PLD001;
Date       03/23/99;
Revision   0;
Designer   Ralph Gibson;
Company    Advanced Micro Devices;
Assembly   UltraNAND(TM) ISA Development Board Interface Solution;
Location   U1;

/*************************************************************************\
**  This is a simple interface for UltraNAND allowing one UltraNAND to  **
**  be supported on a PC ISA bus. This device includes a RESET input to **
**  force WP# asserted on power transitions.  RESET is high until Vcc   **
**  is valid and goes high when supply power ramps down.                **
*************************************************************************
** Allowable Target Device Types:                      AmPALCE16V8-10  **
\*************************************************************************/

ORDER: RESET, %1, !WRITE, %1, !READ, %1, !CE, %1,
       A3, %1, A2, %1, A1, %1, A0, %1, INIT, %1, RY_BY, %2,

       CLE, %1, ALE, %1, !SE, %1, !WP, %1, !OUTCE, %1,
       !WE, %1, !RE, %1, READY;


VECTORS:

$msg    "                                                    ";
$msg    "          !                                 !       ";
$msg    "       R  W  !                 R            O      R ";
$msg    "       E  R  R              I  Y            U      E ";
$msg    "       S  I  E  !           N  -    C  A  !  !  T  !  !  A ";
$msg    "       E  T  A  C  A  A  A  A  I  B    L  L  S  W  C  W  R  D ";
$msg    "       T  E  D  E  3  2  1  0  T  Y    E  E  E  P  E  E  E  Y ";
$msg    "       -  -  -  -  -  -  -  -  -  -    -  -  -  -  -  -  -  - ";
 /*   V0001 */  1  1  1  1  0  0  0  0  0  1    L  L  L  L  H  H  H  Z  /* Reset the device    */
 /*   V0002 */  1  1  1  1  0  0  0  0  0  1    L  L  L  L  H  H  H  Z  /* Reset the device    */
 /*   V0003 */  0  1  1  0  0  0  0  0  0  1    L  L  L  L  H  H  H  Z  /* no function        */
 /*   V0004 */  0  1  1  0  0  0  0  1  0  1    H  L  L  L  H  H  H  Z  /* no function        */
 /*   V0005 */  0  1  1  0  1  0  0  0  0  1    L  L  L  L  H  H  H  Z  /* no function        */
 /*   V0006 */  0  0  1  0  1  0  0  0  0  1    L  L  L  L  L  H  H  Z  /* Set OUTCE#         */
 /*   V0007 */  0  1  1  0  1  0  0  0  0  1    L  L  L  L  L  H  H  Z  /* idle               */
 /*   V0008 */  0  0  1  0  0  0  0  1  0  1    H  L  L  L  L  L  H  Z  /* Write command - CLE */
 /*   V0009 */  0  1  1  0  0  0  0  1  0  1    H  L  L  L  L  H  H  Z  /* idle               */
 /*   V0010 */  0  0  1  0  0  0  1  0  0  1    L  H  L  L  L  H  H  Z  /* Set ALE            */
 /*   V0011 */  0  1  1  0  0  0  1  0  0  1    L  H  L  L  L  H  H  Z  /* idle               */
 /*   V0012 */  0  0  1  0  0  0  0  0  0  1    L  H  L  L  L  L  H  Z  /* Write address      */
 /*   V0013 */  0  1  1  0  0  0  0  0  0  1    L  H  L  L  L  H  H  Z  /* idle               */
 /*   V0014 */  0  0  1  0  0  0  0  0  0  1    L  H  L  L  L  L  H  Z  /* Write address      */
 /*   V0015 */  0  1  1  0  0  0  0  0  0  1    L  H  L  L  L  H  H  Z  /* idle               */
```

```
/*  V0016 */  0  0  1  0  0  0  0  0  0  1    L  H  L  L  L  L  H  Z /* Write address       */
/*  V0017 */  0  1  1  0  0  0  0  0  0  1    L  H  L  L  L  H  H  Z /* idle                */
/*  V0018 */  0  0  1  0  0  0  0  0  0  1    L  H  L  L  L  L  H  Z /* Write address       */
/*  V0019 */  0  1  1  0  0  0  0  0  0  1    L  H  L  L  L  H  H  Z /* idle                */
/*  V0020 */  0  0  1  0  0  0  1  1  0  1    L  L  L  L  L  H  H  Z /* Clear ALE           */
/*  V0021 */  0  1  1  0  0  0  1  1  0  1    L  L  L  L  L  H  H  Z /* idle                */
/*  V0022 */  0  0  1  0  0  1  0  1  0  1    L  L  H  L  L  H  H  Z /* Clear SE#           */
/*  V0023 */  0  1  1  0  0  1  0  1  0  1    L  L  H  L  L  H  H  Z /* idle                */
/*  V0024 */  0  0  1  0  0  1  1  1  0  1    L  L  H  H  L  H  H  Z /* Clear WP#           */
/*  V0025 */  0  1  1  0  0  1  1  1  0  1    L  L  H  H  L  H  H  Z /* idle                */
/*  V0026 */  0  0  1  0  0  0  0  0  0  1    L  L  H  H  L  L  H  Z /* Write data          */
/*  V0027 */  0  1  1  0  0  0  0  0  0  1    L  L  H  H  L  H  H  Z /* idle                */
/*  V0028 */  0  0  1  0  0  0  0  0  0  1    L  L  H  H  L  L  H  Z /* Write data          */
/*  V0029 */  0  1  1  0  0  0  0  0  0  1    L  L  H  H  L  H  H  Z /* idle                */
/*  V0030 */  0  0  1  0  0  1  1  0  0  1    L  L  H  L  L  H  H  Z /* Set WP#             */
/*  V0031 */  0  1  1  0  0  1  1  0  0  1    L  L  H  L  L  H  H  Z /* idle                */
/*  V0032 */  0  0  1  0  0  0  0  1  0  1    H  L  H  L  L  L  H  Z /* Write command - CLE */
/*  V0033 */  0  1  1  0  0  0  0  1  0  1    H  L  H  L  L  H  H  Z /* idle                */
/*  V0034 */  0  0  1  0  0  0  1  0  0  1    L  H  H  L  L  H  H  Z /* Set ALE             */
/*  V0035 */  0  1  1  0  0  0  1  0  0  1    L  H  H  L  L  H  H  Z /* idle                */
/*  V0036 */  0  0  1  0  0  0  0  0  0  1    L  H  H  L  L  L  H  Z /* Write address       */
/*  V0037 */  0  1  1  0  0  0  0  0  0  1    L  H  H  L  L  H  H  Z /* idle                */
/*  V0038 */  0  0  1  0  0  0  0  0  0  1    L  H  H  L  L  L  H  Z /* Write address       */
/*  V0039 */  0  1  1  0  0  0  0  0  0  1    L  H  H  L  L  H  H  Z /* idle                */
/*  V0040 */  0  0  1  0  0  0  0  0  0  1    L  H  H  L  L  L  H  Z /* Write address       */
/*  V0041 */  0  1  1  0  0  0  0  0  0  0    L  H  H  L  L  H  H  Z /* idle                */
/*  V0042 */  0  0  1  0  0  0  1  1  0  0    L  L  H  L  L  H  H  Z /* Clear ALE           */
/*  V0043 */  0  1  1  0  0  0  1  1  0  0    L  L  H  L  L  H  H  Z /* idle                */
/*  V0044 */  0  0  1  0  0  1  0  0  0  0    L  L  L  L  L  H  H  Z /* Set SE#             */
/*  V0045 */  0  1  1  0  0  1  0  0  0  0    L  L  L  L  L  H  H  Z /* idle                */
/*  V0046 */  0  1  0  0  1  1  1  1  0  0    L  L  L  L  L  H  H  L /* Read RY/BY# status  */
/*  V0047 */  0  1  1  0  1  1  1  1  0  0    L  L  L  L  L  H  H  Z /* idle                */
/*  V0048 */  0  1  0  0  1  1  1  1  0  0    L  L  L  L  L  H  H  L /* Read RY/BY# status  */
/*  V0049 */  0  1  1  0  1  1  1  1  0  0    L  L  L  L  L  H  H  Z /* idle                */
/*  V0050 */  0  1  0  0  1  1  1  1  0  0    L  L  L  L  L  H  H  L /* Read RY/BY# status  */
/*  V0051 */  0  1  1  0  1  1  1  1  0  1    L  L  L  L  L  H  H  Z /* idle                */
/*  V0052 */  0  1  0  0  1  1  1  1  0  1    L  L  L  L  L  H  H  H /* Read RY/BY# status  */
/*  V0053 */  0  1  1  0  1  1  1  1  0  1    L  L  L  L  L  H  H  Z /* idle                */
/*  V0054 */  0  1  0  0  0  0  0  0  0  1    L  L  L  L  L  H  L  Z /* Read data           */
/*  V0055 */  0  1  1  0  0  0  0  0  0  1    L  L  L  L  L  H  H  Z /* idle                */
/*  V0056 */  0  1  0  0  0  0  0  0  0  1    L  L  L  L  L  H  L  Z /* Read data           */
/*  V0057 */  0  1  1  0  0  0  0  0  0  1    L  L  L  L  L  H  H  Z /* idle                */
/*  V0058 */  0  0  1  0  0  1  0  1  0  1    L  L  H  L  L  H  H  Z /* Clear SE#           */
/*  V0059 */  0  1  1  0  0  1  0  1  0  1    L  L  H  L  L  H  H  Z /* idle                */
/*  V0060 */  0  0  1  0  0  1  0  0  0  1    L  L  L  L  L  H  H  Z /* Set SE#             */
/*  V0061 */  0  1  1  0  0  1  0  0  0  1    L  L  L  L  L  H  H  Z /* idle                */
/*  V0062 */  0  1  1  0  1  1  1  1  1  1    L  L  L  L  L  H  H  Z /* Initialize          */
/*  V0063 */  0  0  1  1  X  X  0  1  1  1    H  L  L  L  L  L  H  Z /* Write command - CLE */
/*  V0064 */  0  1  1  1  X  X  0  1  1  1    H  L  L  L  L  H  H  Z /* idle                */
/*  V0065 */  0  0  1  1  X  X  1  0  1  1    L  H  L  L  L  H  H  Z /* Set ALE             */
/*  V0066 */  0  1  1  1  X  X  1  0  1  1    L  H  L  L  L  H  H  Z /* idle                */
/*  V0067 */  0  0  1  1  X  X  0  0  1  1    L  H  L  L  L  L  H  Z /* Write address       */
/*  V0068 */  0  1  1  1  X  X  0  0  1  1    L  H  L  L  L  H  H  Z /* idle                */
```

```
/*  V0069 */  0  0  1  1  X  X  0  0  1  1    L  H  L  L  L  L  H  Z  /* Write address      */
/*  V0070 */  0  1  1  1  X  X  0  0  1  1    L  H  L  L  L  H  H  Z  /* idle               */
/*  V0071 */  0  0  1  1  X  X  0  0  1  1    L  H  L  L  L  L  H  Z  /* Write address      */
/*  V0072 */  0  1  1  1  X  X  0  0  1  1    L  H  L  L  L  H  H  Z  /* idle               */
/*  V0073 */  0  0  1  1  X  X  1  1  1  1    L  L  L  L  L  H  H  Z  /* Clear ALE          */
/*  V0074 */  0  1  1  1  X  X  1  1  1  1    L  L  L  L  L  H  H  Z  /* idle               */
/*  V0075 */  0  1  0  1  X  X  0  0  1  1    L  L  L  L  L  H  L  Z  /* Read data          */
/*  V0076 */  0  1  1  1  X  X  0  0  1  1    L  L  L  L  L  H  H  Z  /* idle               */
/*  V0077 */  0  1  0  1  X  X  0  0  1  1    L  L  L  L  L  H  L  Z  /* Read data          */
/*  V0078 */  0  1  1  1  X  X  0  0  1  1    L  L  L  L  L  H  H  Z  /* idle               */
/*  V0079 */  1  1  1  0  1  1  1  1  0  1    L  L  L  L  H  H  H  Z  /* Reset the device   */
/*  V0080 */  1  1  1  1  1  1  1  1  0  1    L  L  L  L  H  H  H  Z  /* idle               */
/*  V0081 */  0  0  1  0  1  0  0  0  0  1    L  L  L  L  L  H  H  Z  /* Set OUTCE#         */
/*  V0082 */  0  1  1  0  1  0  0  0  0  1    L  L  L  L  L  H  H  Z  /* idle               */
/*  V0083 */  0  1  1  1  1  1  1  1  0  1    L  L  L  L  L  H  H  Z  /* idle               */
/*  V0084 */  0  0  1  0  1  0  0  1  0  1    L  L  L  L  H  H  H  Z  /* Clear OUTCE#       */
/*  V0085 */  0  1  1  0  1  0  0  1  0  1    L  L  L  L  H  H  H  Z  /* idle               */
/*  V0086 */  0  1  1  1  1  1  1  1  0  1    L  L  L  L  H  H  H  Z  /* idle               */
```



Note:
The AmPALLV16V8-10SC is a 3.3 Volt PLD device.

Figure 4. Example PLD Implementation (One Bank Support)

PLD Source Code for UNISA22

This example includes the PLD source code and the simulation file used to verify a slightly modified PLD design. The PLD circuit included below supports up to three UltraNAND banks. The AMD AmPALLV22V10-10PC used in the example powers up in a known state which covers initial power-up conditions. The RESET input is used to initialize the transparent latches in the PLD. Figure 5 is a diagram showing the pin-out of the AmPALLV22V10-10PC used in this design example.

```
Name         UNISA22;
Partno       PLD001;
Date         03/23/99;
Revision     0;
Designer     Ralph Gibson;
Company      Advanced Micro Devices;
Assembly     UltraNAND(TM) ISA Development Board Interface Solution;
Location     U1;
Device       P22V10;
Format       J;

/*****************************************************************************\
**  This is a simple interface for UltraNAND allowing one UltraNAND to be   **
**  supported on a PC ISA bus. This device includes a RESET input to force  **
**  WP# asserted on power transitions.  RESET is high until Vcc is valid    **
**  and goes high when supply power ramps down.                             **
\*****************************************************************************/

/**  Inputs  **/

Pin 1        = !WRITE;                              /* System Write Enable */
Pin 2        = !READ;                               /* System Read Enable */
Pin [3..6]   = [A0..3];                    /* Address input to select port */
Pin 7        = !CE;          /* Chip Enable for the interface and UltraNAND */
Pin 8        = RY_BY;                       /* RY/BY# input from UltraNAND */
Pin 9        = RESET;       /* RESET - high for reset and power transitions */
Pin 10       = INIT;              /* INIT from the optional Boot Loader PLD */

/**  Outputs  **/

Pin 23       = READY;             /* Allows system to read RY/BY# pin state */
Pin 22       = CLE;                    /* Command Latch Enable to UltraNAND */
Pin 21       = ALE;                    /* Address Latch Enable to UltraNAND */
Pin 20       = !SE;                       /* Spare Area Enable to UltraNAND */
Pin 19       = !WP;                           /* Write Protect to UltraNAND */
Pin [18..16] = [!OUTCE0..2];                  /* Chip Enables to UltraNAND */
Pin 15       = !WE;                           /* Write Enable to UltraNAND */
Pin 14       = !RE;                            /* Read Enable to UltraNAND */


/** Declarations and Intermediate Variable Definitions **/

PORT0 =    CE & !A3 & !A2  & !A1 & !A0             /* Data read/write port */
      #  INIT & !A1 & !A0;
PORT1 =    CE & !A3 & !A2  & !A1 &  A0                   /* CLE write port */
      #  INIT & !A1 &  A0;
PORT2 =    CE & !A3 & !A2  &  A1 & !A0                  /* Used to set ALE */
      #  INIT &  A1 & !A0;
```

```
PORT3 =    CE & !A3 & !A2 &  A1 &  A0                  /* Used to clear ALE */
      #  INIT &  A1 &  A0;
PORT4 =    CE & !A3 &  A2 & !A1 & !A0;                   /* Used to set SE# */
PORT5 =    CE & !A3 &  A2 & !A1 &  A0;                  /* Used to clear SE# */
PORT6 =    CE & !A3 &  A2 &  A1 & !A0;                   /* Used to set WP# */
PORT7 =    CE & !A3 &  A2 &  A1 &  A0;                  /* Used to clear WP# */
PORT8 =    CE &  A3 & !A2 & !A1 & !A0;                 /* Used to set OUTCE# */
PORT9 =    CE &  A3 & !A2 & !A1 &  A0;               /* Used to clear OUTCE# */
PORTA =    CE &  A3 & !A2 &  A1 & !A0;                        /* No Function */
PORTB =    CE &  A3 & !A2 &  A1 &  A0;                        /* No Function */
PORTC =    CE &  A3 &  A2 & !A1 & !A0;                        /* No Function */
PORTD =    CE &  A3 &  A2 & !A1 &  A0;                        /* No Function */
PORTE =    CE &  A3 &  A2 &  A1 & !A0;                        /* No Function */
PORTF =    CE &  A3 &  A2 &  A1 &  A0;               /* To read RY/BY# state */

/**  Logic Equations  **/

READY.OE = PORTF & READ;        /* READY is only driven during a PORTF read */

READY    = RY_BY; /* READY shows the state of RY/BY# */

CLE      = PORT1; /* Assert CLE on all PORT1 accesses */

ALE      = !RESET & ( WRITE & PORT2          /* Latch ALE on write to PORT2 */
                  # ALE & !( WRITE & PORT3 ) /* and clear on write to PORT3 */
                  # ALE & PORT2 );          /* Transparent latch cover term */

SE       = RESET                             /* Latch SE# asserted on RESET */
         # WRITE & PORT4                     /* Latch SE# on write to PORT4 */
         # SE & !( WRITE & PORT5 )           /* and clear on write to PORT5 */
         # SE & PORT4;                      /* Transparent latch cover term */

WP       = RESET                             /* Latch WP# asserted on RESET */
         # WRITE & PORT6                     /* Latch WP# on write to PORT6 */
         # WP & !( WRITE & PORT7 )           /* and clear on write to PORT7 */
         # WP & PORT6;                      /* Transparent latch cover term */

OUTCE0   = !RESET & ( INIT                  /* Latch OUTCE# asserted on INIT */
                  # WRITE & PORT8     /* Latch OUTCE# on write to PORT2 and */
                  # OUTCE0 & !( WRITE & PORT9 )  /* clear on write to PORT3 */
                  # OUTCE0 & PORT8 );       /* Transparent latch cover term */

OUTCE1   = !RESET & ( INIT                  /* Latch OUTCE# asserted on INIT */
                  # WRITE & PORTA     /* Latch OUTCE# on write to PORT2 and */
                  # OUTCE1 & !( WRITE & PORTB )  /* clear on write to PORT3 */
                  # OUTCE1 & PORTA );       /* Transparent latch cover term */

OUTCE2   = !RESET & ( INIT                  /* Latch OUTCE# asserted on INIT */
                  # WRITE & PORTC     /* Latch OUTCE# on write to PORT2 and */
                  # OUTCE2 & !( WRITE & PORTD )  /* clear on write to PORT3 */
                  # OUTCE2 & PORTC );       /* Transparent latch cover term */

WE       = WRITE & ( PORT0     /* Drive WE# to UltraNAND for PORT0 or PORT1 */
                 # PORT1 );

RE       = READ & PORT0;           /* Drive RE# to UltraNAND for PORT0 only */
```

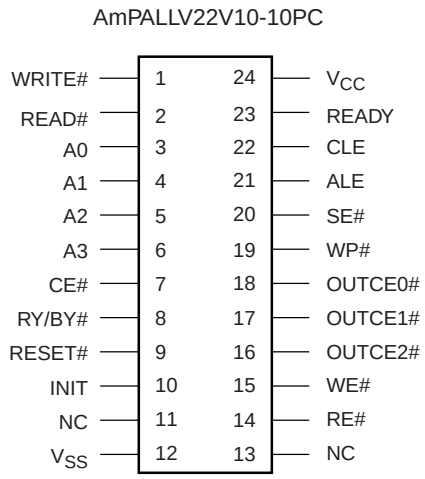
PLD Simulation File for UNISA22

The simulation file included here is used to verify the design of the PLD supporting up to three UltraNAND banks.

```
Name       UNISA22;
Partno     PLD001;
Date       03/23/99;
Revision   0;
Designer   Ralph Gibson;
Company    Advanced Micro Devices;
Assembly   UltraNAND(TM) ISA Development Board Interface Solution;
Location   U1;

/*************************************************************************\
**  This is a simple interface for UltraNAND allowing one UltraNAND to be   **
**  supported on a PC ISA bus. This device includes a RESET input to force  **
**  WP# asserted on power transitions.  RESET is high until Vcc is valid    **
**  and goes high when supply power ramps down.                             **
*************************************************************************
** Allowable Target Device Types:                      AmPALCE22V10-10  **
\*************************************************************************/

ORDER: RESET, %1, !WRITE, %1, !READ, %1, !CE, %1,
       A3, %1, A2, %1, A1, %1, A0, %1, INIT, %1, RY_BY, %2,

       CLE, %1, ALE, %1, !SE, %1, !WP, %1,
       !OUTCE0, %1, !OUTCE1, %1, !OUTCE2, %1, !WE, %1, !RE, %1, READY;


VECTORS:

$msg   "                                          ! ! !        ";
$msg   "         !                                O O O        ";
$msg   "      R W !              R               U U U      R ";
$msg   "      E R R            I Y               T T T      E ";
$msg   "      S I E !          N -    C A ! ! C C C ! ! A ";
$msg   "      E T A C A A A A I B    L L S W E E E W R D ";
$msg   "      T E D E 3 2 1 0 T Y    E E E P 0 1 2 E E Y ";
$msg   "      - - - - - - - - - -    - - - - -      - - -
 /*  V0001 */ 1 1 1 1 0 0 0 0 0 1    L L L L H H H H H Z /* Reset the device  */
 /*  V0002 */ 1 1 1 1 0 0 0 0 0 1    L L L L H H H H H Z /* Reset the device  */
 /*  V0003 */ 0 1 1 0 0 0 0 0 0 1    L L L L H H H H H Z /* no function       */
 /*  V0004 */ 0 1 1 0 0 0 0 1 0 1    H L L L H H H H H Z /* no function       */
 /*  V0005 */ 0 1 1 0 1 0 0 0 0 1    L L L L H H H H H Z /* no function       */
 /*  V0006 */ 0 0 1 0 1 0 0 0 0 1    L L L L L H H H H Z /* Set OUTCE0#       */
 /*  V0007 */ 0 1 1 0 1 0 0 0 0 1    L L L L L H H H H Z /* idle              */
 /*  V0008 */ 0 0 1 0 1 0 1 0 0 1    L L L L L L H H H Z /* Set OUTCE1#       */
 /*  V0009 */ 0 1 1 0 1 0 1 0 0 1    L L L L L L H H H Z /* idle              */
 /*  V0010 */ 0 0 1 0 1 1 0 0 0 1    L L L L L L L H H Z /* Set OUTCE2#       */
 /*  V0011 */ 0 1 1 0 1 1 0 0 0 1    L L L L L L L H H Z /* idle              */
 /*  V0012 */ 0 0 1 0 0 0 0 1 0 1    H L L L L L L L H Z /* Write command-CLE */
 /*  V0013 */ 0 1 1 0 0 0 0 1 0 1    H L L L L L L H H Z /* idle              */
 /*  V0014 */ 0 0 1 0 0 0 1 0 0 1    L H L L L L L H H Z /* Set ALE           */
```

```
/*  V0015 */ 0 1 1 0 0 0 1 0 0 1   L H L L L L L H H Z /* idle               */
/*  V0016 */ 0 0 1 0 0 0 0 0 0 1   L H L L L L L L H Z /* Write address      */
/*  V0017 */ 0 1 1 0 0 0 0 0 0 1   L H L L L L L H H Z /* idle               */
/*  V0018 */ 0 0 1 0 0 0 0 0 0 1   L H L L L L L L H Z /* Write address      */
/*  V0019 */ 0 1 1 0 0 0 0 0 0 1   L H L L L L L H H Z /* idle               */
/*  V0020 */ 0 0 1 0 0 0 0 0 0 1   L H L L L L L L H Z /* Write address      */
/*  V0021 */ 0 1 1 0 0 0 0 0 0 1   L H L L L L L H H Z /* idle               */
/*  V0022 */ 0 0 1 0 0 0 0 0 0 1   L H L L L L L L H Z /* Write address      */
/*  V0023 */ 0 1 1 0 0 0 0 0 0 1   L H L L L L L H H Z /* idle               */
/*  V0024 */ 0 0 1 0 0 0 1 1 0 1   L L L L L L L H H Z /* Clear ALE          */
/*  V0025 */ 0 1 1 0 0 0 1 1 0 1   L L L L L L L H H Z /* idle               */
/*  V0026 */ 0 0 1 0 0 1 0 1 0 1   L L H L L L L H H Z /* Clear SE#          */
/*  V0027 */ 0 1 1 0 0 1 0 1 0 1   L L H L L L L H H Z /* idle               */
/*  V0028 */ 0 0 1 0 0 1 1 1 0 1   L L H H L L L H H Z /* Clear WP#          */
/*  V0029 */ 0 1 1 0 0 1 1 1 0 1   L L H H L L L H H Z /* idle               */
/*  V0030 */ 0 0 1 0 0 0 0 0 0 1   L L H H L L L L H Z /* Write data         */
/*  V0031 */ 0 1 1 0 0 0 0 0 0 1   L L H H L L L H H Z /* idle               */
/*  V0032 */ 0 0 1 0 0 0 0 0 0 1   L L H H L L L L H Z /* Write data         */
/*  V0033 */ 0 1 1 0 0 0 0 0 0 1   L L H H L L L H H Z /* idle               */
/*  V0034 */ 0 0 1 0 0 1 1 0 0 1   L L H L L L L H H Z /* Set WP#            */
/*  V0035 */ 0 1 1 0 0 1 1 0 0 1   L L H L L L L H H Z /* idle               */
/*  V0036 */ 0 0 1 0 0 0 0 1 0 1   H L H L L L L L H Z /* Write command-CLE  */
/*  V0037 */ 0 1 1 0 0 0 0 1 0 1   H L H L L L L H H Z /* idle               */
/*  V0038 */ 0 0 1 0 0 0 1 0 0 1   L H H L L L L H H Z /* Set ALE            */
/*  V0039 */ 0 1 1 0 0 0 1 0 0 1   L H H L L L L H H Z /* idle               */
/*  V0040 */ 0 0 1 0 0 0 0 0 0 1   L H H L L L L L H Z /* Write address      */
/*  V0041 */ 0 1 1 0 0 0 0 0 0 1   L H H L L L L H H Z /* idle               */
/*  V0042 */ 0 0 1 0 0 0 0 0 0 1   L H H L L L L L H Z /* Write address      */
/*  V0043 */ 0 1 1 0 0 0 0 0 0 1   L H H L L L L H H Z /* idle               */
/*  V0044 */ 0 0 1 0 0 0 0 0 0 1   L H H L L L L L H Z /* Write address      */
/*  V0045 */ 0 1 1 0 0 0 0 0 0 0   L H H L L L L H H Z /* idle               */
/*  V0046 */ 0 0 1 0 0 0 1 1 0 0   L L H L L L L H H Z /* Clear ALE          */
/*  V0047 */ 0 1 1 0 0 0 1 1 0 0   L L H L L L L H H Z /* idle               */
/*  V0048 */ 0 0 1 0 0 1 0 0 0 0   L L L L L L L H H Z /* Set SE#            */
/*  V0049 */ 0 1 1 0 0 1 0 0 0 0   L L L L L L L H H Z /* idle               */
/*  V0050 */ 0 1 0 0 1 1 1 1 0 0   L L L L L L L H H L /* Read RY/BY# status */
/*  V0051 */ 0 1 1 0 1 1 1 1 0 0   L L L L L L L H H Z /* idle               */
/*  V0052 */ 0 1 0 0 1 1 1 1 0 0   L L L L L L L H H L /* Read RY/BY# status */
/*  V0053 */ 0 1 1 0 1 1 1 1 0 0   L L L L L L L H H Z /* idle               */
/*  V0054 */ 0 1 0 0 1 1 1 1 0 0   L L L L L L L H H L /* Read RY/BY# status */
/*  V0055 */ 0 1 1 0 1 1 1 1 0 1   L L L L L L L H H Z /* idle               */
/*  V0056 */ 0 1 0 0 1 1 1 1 0 1   L L L L L L L H H H /* Read RY/BY# status */
/*  V0057 */ 0 1 1 0 1 1 1 1 0 1   L L L L L L L H H Z /* idle               */
/*  V0058 */ 0 1 0 0 0 0 0 0 0 1   L L L L L L L H L Z /* Read data          */
/*  V0059 */ 0 1 1 0 0 0 0 0 0 1   L L L L L L L H H Z /* idle               */
/*  V0060 */ 0 1 0 0 0 0 0 0 0 1   L L L L L L L H L Z /* Read data          */
/*  V0061 */ 0 1 1 0 0 0 0 0 0 1   L L L L L L L H H Z /* idle               */
/*  V0062 */ 0 0 1 0 0 1 0 1 0 1   L L H L L L L H H Z /* Clear SE#          */
/*  V0063 */ 0 1 1 0 0 1 0 1 0 1   L L H L L L L H H Z /* idle               */
```

```
/*  V0064 */ 0 0 1 0 0 1 0 0 0 1   L L L L L L L H H Z /* Set SE#           */
/*  V0065 */ 0 1 1 0 0 1 0 0 0 1   L L L L L L L H H Z /* idle              */
/*  V0066 */ 0 1 1 0 1 1 1 1 1 1   L L L L L L L H H Z /* Initialize        */
/*  V0067 */ 0 0 1 1 X X 0 1 1 1   H L L L L L L L H Z /* Write command-CLE */
/*  V0068 */ 0 1 1 1 X X 0 1 1 1   H L L L L L L H H Z /* idle              */
/*  V0069 */ 0 0 1 1 X X 1 0 1 1   L H L L L L L H H Z /* Set ALE           */
/*  V0070 */ 0 1 1 1 X X 1 0 1 1   L H L L L L L H H Z /* idle              */
/*  V0071 */ 0 0 1 1 X X 0 0 1 1   L H L L L L L L H Z /* Write address     */
/*  V0072 */ 0 1 1 1 X X 0 0 1 1   L H L L L L L H H Z /* idle              */
/*  V0073 */ 0 0 1 1 X X 0 0 1 1   L H L L L L L L H Z /* Write address     */
/*  V0074 */ 0 1 1 1 X X 0 0 1 1   L H L L L L L H H Z /* idle              */
/*  V0075 */ 0 0 1 1 X X 0 0 1 1   L H L L L L L L H Z /* Write address     */
/*  V0076 */ 0 1 1 1 X X 0 0 1 1   L H L L L L L H H Z /* idle              */
/*  V0077 */ 0 0 1 1 X X 1 1 1 1   L L L L L L L H H Z /* Clear ALE         */
/*  V0078 */ 0 1 1 1 X X 1 1 1 1   L L L L L L L H H Z /* idle              */
/*  V0079 */ 0 1 0 1 X X 0 0 1 1   L L L L L L L H L Z /* Read data         */
/*  V0080 */ 0 1 1 1 X X 0 0 1 1   L L L L L L L H H Z /* idle              */
/*  V0081 */ 0 1 0 1 X X 0 0 1 1   L L L L L L L H L Z /* Read data         */
/*  V0082 */ 0 1 1 1 X X 0 0 1 1   L L L L L L L H H Z /* idle              */
/*  V0083 */ 1 1 1 0 1 1 1 1 0 1   L L L L H H H H H Z /* Reset the device  */
/*  V0084 */ 1 1 1 1 1 1 1 1 0 1   L L L L H H H H H Z /* idle              */
/*  V0085 */ 0 0 1 0 1 0 0 0 0 1   L L L L L H H H H Z /* Set OUTCE0#       */
/*  V0086 */ 0 1 1 0 1 0 0 0 0 1   L L L L L H H H H Z /* idle              */
/*  V0087 */ 0 0 1 0 1 0 1 0 0 1   L L L L L L H H H Z /* Set OUTCE1#       */
/*  V0088 */ 0 1 1 0 1 0 1 0 0 1   L L L L L L H H H Z /* idle              */
/*  V0089 */ 0 0 1 0 1 1 0 0 0 1   L L L L L L L H H Z /* Set OUTCE2#       */
/*  V0090 */ 0 1 1 0 1 1 0 0 0 1   L L L L L L L H H Z /* idle              */
/*  V0091 */ 0 1 1 1 1 1 1 1 0 1   L L L L L L L H H Z /* idle              */
/*  V0092 */ 0 0 1 0 1 0 0 1 0 1   L L L L H L L H H Z /* Clear OUTCE0#     */
/*  V0093 */ 0 1 1 0 1 0 0 1 0 1   L L L L H L L H H Z /* idle              */
/*  V0094 */ 0 0 1 0 1 0 1 1 0 1   L L L L H H L H H Z /* Clear OUTCE1#     */
/*  V0095 */ 0 1 1 0 1 0 1 1 0 1   L L L L H H L H H Z /* idle              */
/*  V0096 */ 0 0 1 0 1 1 0 1 0 1   L L L L H H H H H Z /* Clear OUTCE2#     */
/*  V0097 */ 0 1 1 0 1 1 0 1 0 1   L L L L H H H H H Z /* idle              */
/*  V0098 */ 0 1 1 1 1 1 1 1 0 1   L L L L H H H H H Z /* idle              */
```



Note:
The AmPALLV22V10-10PC is a 3.3 Volt PLD device.

Figure 5. Example PLD Implementation (Three Bank Support)

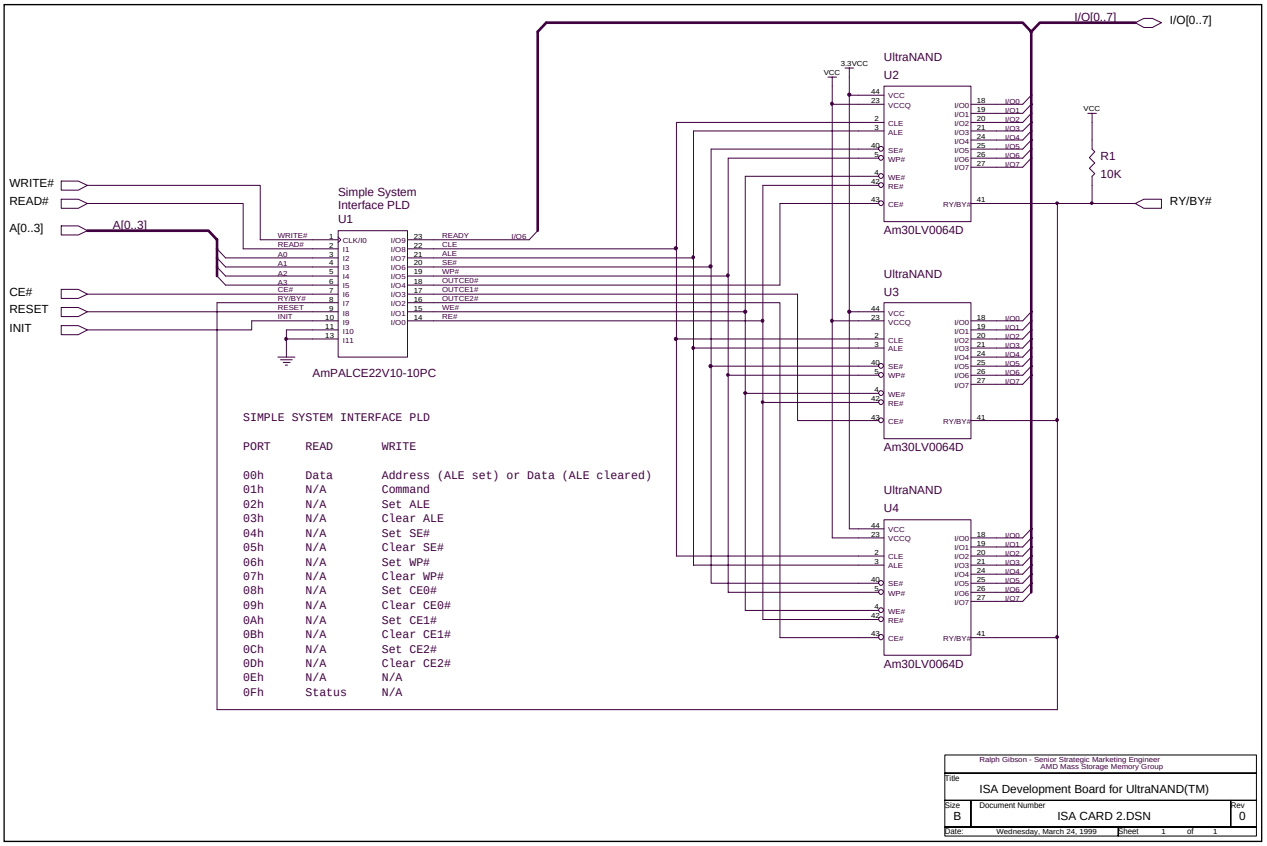


Figure 6. Schematic Diagram for PLD with Support for Three UltraNAND Banks

One AMD Place
P.O. Box 3453
Sunnyvale,
California 94088-3453
USA
(408) 732-2400
(800) 538-8450
TWX: 910-339-9280
TELEX: 34-6306

TECHNICAL SUPPORT
USA & CANADA non-PC CPU: (800) 222-9323
USA & CANADA PC CPU: (408) 749-3060
USA & CANADA PC CPU E-mail: hw.support@amd.com
EUROPE & UK: +44-(0)1276 803299
Fax: +44-(0)1276 803298
BBS: +44-(0)1276 803211
FRANCE: 0800 90 8621
GERMANY: 089-450-53199
ITALY: 1678-77224
EUROPE E-mail: euro.tech@amd.com
FAR EAST Fax: (852) 2956-0599
JAPAN: 03-3346-7550
Fax: 03-3346-7848
ARGENTINA: 001-800-200-1111,
after tone 800-859-4478
CHILE: 800-532-853
MEXICO: 95-800-222-9323

LITERATURE ORDERING
USA & CANADA: (800) 222-9323
EUROPE E-mail: euro.lit@amd.com
FAR EAST Fax: (852) 2956-0599
JAPAN Fax: 03-3346-9628

www.amd.com

Printed in USA
4/XX/99
22363A