

## Introduction

Storage devices, such as the floppy drive, CD-ROM drive, and hard disk drive, are connected to the computer through the IDE/ATA interface. This design example illustrates the implementation of an IDE/ATA controller using a MAX® II CPLD through which a host computer or microprocessor system can connect to a standard Integrated Drive Electronics (IDE) device.

## IDE/ATA Controller and Interface

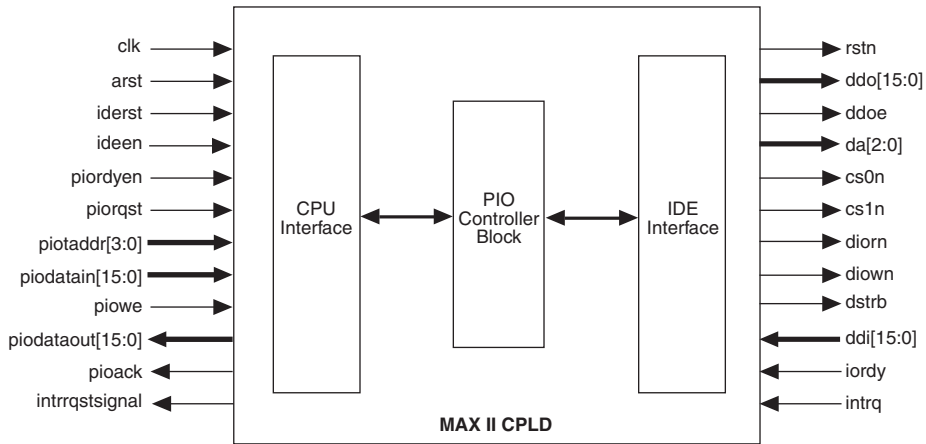
When controllers and hard drives had proprietary technologies, a controller from one manufacturer did not work well with a hard drive from another manufacturer. The IDE was created to standardize the use of hard drives in computers. This was based on a concept of combining the controller and the hard drive, thereby reducing interface costs and making firmware implementations easier. The controller residing on a chip provided the means for transferring data to or from the host computer.

This IDE controller, also known as the ATA (Advanced Technology Attachment) controller, is an asynchronous parallel interface between a host microprocessor system and a standard IDE device. Therefore, this can be called a host adapter because it provides a way to connect a complete IDE device to the host.

From the time of its inception, the ATA interface has been upgraded frequently and newer versions have been introduced. This design example implements an IDE controller compatible with the ATA-5 interface. Although the ATA-5 standard supports two modes of operation—the PIO mode and the DMA mode—this design is restricted to only the PIO mode (mode 0) and with only one device connected to the controller (master).

Figure 1 shows the block diagram of an IDE/ATA interface.

Figure 1. Basic Block Diagram of an IDE/ATA Interface Note (1)



Note to Figure 1:

- (1) The IDE/ATA interface consists of three blocks: the CPU interface block that receives commands from the CPU, the PIO controller block that contains the PIO state machine, and the IDE interface block that generates the signals required by the IDE device to carry out data transfer with the host (computer).

## IDE/ATA Controller using MAX II CPLD

The IDE Interface supports two modes of data transfer: the PIO mode and the DMA mode. This design example is restricted to mode 0 of PIO data transfer.



For more information about the different PIO modes and DMA modes, refer to the ATA/ATAPI-5 specification available at: [www.t13.org/Documents/UploadedDocuments/project/d1321r3-ATA-ATAPI-5.pdf](http://www.t13.org/Documents/UploadedDocuments/project/d1321r3-ATA-ATAPI-5.pdf)

Table 1 provides a brief description of the various signals shown in Figure 1 on page 2.

<b>Signal</b>	<b>Size</b>	<b>Direction</b>	<b>Description</b>
clk	1	Input	Same as the clock of the processor. This example works at a clock frequency of 100 MHz.
arst	1	Input	Asynchronous active-low reset to reset the controller.
iderst	1	Input	Active-high signal to reset the IDE device.
ideen	1	Input	Active-high signal to enable the IDE device.
pioiordyen	1	Input	Active-high signal to enable the IORDY signal from the IDE device.
piorqst	1	Input	Active-high signal to start a PIO data transfer cycle.
pioaddr[3:0]	4	Input	4-bit bus to select the device address and the chip select signals of the IDE device.
piodatain[15:0]	16	Input	16-bit bus to send the data to the IDE device.
piowe	1	Input	Active-high signal to set the direction of data transfer. piowe = 1; write operation piowe = 0; read operation
intrrqstsignal	1	Output	Signal to interrupt the CPU.
pioack	1	Output	Signal to indicate the end of a PIO read/write cycle.
piodataout [15:0]	16	Output	16-bit bus to hold the data read from the IDE device.
rstn	1	Output	Active-low signal to reset the IDE device.
ddo [15:0]	16	Output	16-bit data bus that transfers the data sent by the CPU to the device. The lower 8 bits are used for 8-bit data transfers.
da [2:0]	3	Output	3-bit active-high signal. Contains the binary coded address asserted by the host to access a register or data port in the device.
cs0n, cs1n	1 bit each	Output	Active-low chip select signals from the host used to select the command block or control block registers. CS0=0; select control register CS1= 0; select command register CS0, CS1= 0/1; the device will ignore transitions on DIOR-/DIOW- and release the data bus.
diorn	1	Output	Active-low strobe signal asserted by the host to read device registers or the data port.
diown	1	Output	Active-low strobe signal asserted by the host to write to the device registers or the data port.

**Table 1. Description of Interface Signals (Part 2 of 2)**

Signal	Size	Direction	Description
dstrb	1	Output	data-in strobe signal from the device. The rising edge of dstrb latches the data from the device into the host.
ddi [15:0]	16	Input	16-bit data bus that contains the data read from the IDE device.
iordy	1	Input	This signal is negated to extend the host transfer cycle of any host register access (read or write) when the device is not ready to respond to a data transfer request. Optional for mode 0 but required for higher modes.
intrq	1	Input	Used by the selected device to notify the host of an event. The device internal interrupt pending state is set when such an event occurs.
ddoe	1	Output	Used to read data from the IDE device. The data on the ddi bus is put on the data lines piodataout after the signal is brought low.

### CPU Interface Block

The CPU interface block receives the signals from the host CPU and stores them in the internal registers of the CPLD. Depending on the information in these internal registers, the PIO controller goes through the various states of the PIO read and write operations.

### PIO Controller Block

The PIO controller block contains the PIO state machine. Whenever the host sends a read or write request, the state machine goes through the appropriate read or write states, resulting in data transfer. A module called the *piomodecontroller* in this block determines the states through which the state machine should proceed. Another module called the *runoncecounter* loads the count value to generate the required delays as per the read or write timing requirements. The count value is loaded through parameters called *piomodet1*, *piomodet2*, *piomodet4*, and *piomodeteoc*. The default values loaded in this design example are for a processor operating at a frequency of 100 MHz. These parameter values may be modified for a processor operating at a different frequency. A third module called the *updowncounter* decrements the count loaded on each clock pulse, thus generating the required delay.



For more information about the PIO read and write waveforms and timing specifications, refer to the ATA/ATAPI-5 specification available at:  
[www.t13.org/Documents/UploadedDocuments/project/d1321r3-ATA-ATAPI-5.pdf](http://www.t13.org/Documents/UploadedDocuments/project/d1321r3-ATA-ATAPI-5.pdf)

### The IDE Interface Block

This IDE interface block generates the appropriate interfacing signals so that the data is loaded into or from the addressed internal register of the IDE device. The internal register selected depends on the value of the `da[2:0]`, `cs0n`, and `cs1n` lines. A read or write operation is indicated by the `diorn` and the `diown` lines., respectively.



For more information about the various internal registers of the IDE device and methods of selecting them, refer to the ATA/ATAPI-5 specification available at:  
[www.t13.org/Documents/UploadedDocuments/project/d1321r3-ATA-ATAPI-5.pdf](http://www.t13.org/Documents/UploadedDocuments/project/d1321r3-ATA-ATAPI-5.pdf)

## Implementation

This design example can be implemented using a MAX II device such as the EPM570 device or any other MAX II CPLD with the required general-purpose I/O (GPIO) pin count and LEs.

## Source Code

This design example has been implemented in Verilog and successful operation has been demonstrated using the MDN-B2 demo board, as described in this application note. The source code, testbench, and complete Quartus II project are available at:

[www.altera.com/literature/an/an495\\_design\\_example.zip](http://www.altera.com/literature/an/an495_design_example.zip)

## Conclusion

As shown in this design example, MAX II CPLDs can be efficiently utilized in implementing an IDE/ATA controller. With the standardized ATA interface, MAX II CPLDs can be utilized to interface any storage device to a host microcontroller or processor system. The MAX II CPLDs, with their high density I/O counts, low cost, easy power-on sequencing, and multi-volt features make them the perfect device to implement an IDE/ATA controller.

## Additional Resources

- MAX II CPLD Homepage:  
[www.altera.com/products/devices/cpld/max2/mx2-index.jsp](http://www.altera.com/products/devices/cpld/max2/mx2-index.jsp)
- MAX II Device Literature:  
[www.altera.com/literature/lit-max2.jsp](http://www.altera.com/literature/lit-max2.jsp)
- MAX II Power-Down Designs:  
[www.altera.com/support/examples/max/exm-power-down.html](http://www.altera.com/support/examples/max/exm-power-down.html)
- MAX II Application Notes:
  - *AN 422: Power Management in Portable Systems Using MAX II CPLDs*
  - *AN 428: MAX II CPLD Design Guidelines*

## Document Revision History

Table 2 shows the revision history for this application note.

<i>Table 2. Document Revision History</i>		
<b>Date and Document Version</b>	<b>Changes Made</b>	<b>Summary of Changes</b>
December 2007 v1.0	Initial release.	—



101 Innovation Drive  
San Jose, CA 95134  
[www.altera.com](http://www.altera.com)  
**Technical Support:**  
[www.altera.com/support/](http://www.altera.com/support/)  
**Literature Services:**  
[literature@altera.com](mailto:literature@altera.com)

Copyright © 2007 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

