# ATA IDE Controller for the XSB-300E Board

April 14, 2004 (Version 1.0)

Application Note by D. Vanden Bout


## Summary

This application note describes a simple ATA IDE controller core that allows an FPGA to access a hard disk or Compact Flash card on the XSB-300E Board.


## ATA Controller Features

The ATA IDE controller core lets the FPGA access a hard disk or Compact Flash card (in True IDE mode) that is connected to the XSB-300E Board. On the host side, the controller allows the FPGA to read or write an entire 512-byte sector by supplying the sector, cylinder and head parameters. The ATA controller accesses the disk or CF card through a low-level interface that creates read/write pulses conforming to PIO mode 0 timing specifications. The ATA controller generates all the timing and synchronization signals required on both the disk/CF and host sides.

## ATA Controller Parameters and I/O

### *Generic Parameters*

A single generic parameter affects the operation of the ATA controller:

**FREQ**: This parameter sets the operating frequency of the controller. This frequency is used to calculate the appropriate values such that the low-level disk/CF read and write operations meet the PIO mode 0 timing specifications.

### *I/O Ports*

The host- and disk/CF-side connections for the controller are shown in Figure 1. The functions of the I/O signals are as follows:

**clk**: This is the main clock input. The clock from the external oscillator enters the FPGA through a global clock input pin and drives this input.

**rst**: This active-high, asynchronous input resets the internal circuitry of the controller and also causes a reset of the disk or CF card.

**rd**: This active-high input initiates a read of a single word from the disk/CF sector. It is sampled on the rising clock edge and must be held high until the done signal indicates the 16-bit data word is available. The read control must be lowered before the next rising clock edge after the done signal goes high or else the next word in the sector will be read.

**wr**: This active-high input initiates a write of a single word to the disk/CF sector. It is sampled on the rising clock edge and must be held high until the done signal indicates the 16-bit data word from the host has been written to the disk/CF. The write control must be lowered before the next rising clock edge after the done signal goes high or else the next word in the sector will be written.

**abort**: This active-high input terminates the read or write of a disk/CF sector. It is sampled on the rising clock edge. Aborting a write sector command before all 256 words are written leaves the sector in an indeterminate state. Aborting a read sector command before all 256 words are read does not affect the sector contents.

**head**: This four-bit bus supplies the address of the disk head containing the sector on the disk/CF that is to be read or written. The head address must be held stable until the done signal goes high.

**cylinder**: This 16-bit bus supplies the address of the disk cylinder containing the sector on the disk/CF that is to be read or written. The cylinder address must be held stable until the done signal goes high.

**sector**: This eight-bit bus supplies the address of the disk sector on the disk/CF that is to be read or

written. The sector address must be held stable until the done signal goes high.

**hDIn**: The 16-bit data to be written to the disk/CF enters through this input bus. The data value must be held stable until the done signal goes high.

**hDOut**: The 16-bit data read from the disk/CF comes out on this bus. This data must be latched by the host-side logic on the rising clock edge after the done signal goes high.

**done**: This synchronous output signal goes high to indicate the completion of the currently active read or write operation. It remains high for a single clock cycle.

**status**: The current status of the ATA controller controller is made available on this four-bit bus. The least-significant bit will be set if an error occurred during the read/write operation. The ATA controller must be reset to clear the error.

## Using the ATA Controller

The waveforms for reading a disk/CF sector are shown in Figure 2. At ①, the host provides the head, cylinder and sector numbers and drives the read control line high. After this, multiple cycles are needed for the disk/CF to address and access the given sector. At ②, the first data word from the sector is output to the host on the hDOut bus and the done signal is raised. At ③, the host latches the value on hDOut and lowers rd before the next rising clock edge. The host raises the rd input at ④ to get the next data word from the sector. The ATA controller will respond quickly by raising done at ⑤ since the sector contents have already been buffered. Then the host latches the new data and lowers the rd input at ⑥. This procedure is repeated an additional 254 times to read the contents of the entire 256-word sector.

The waveforms for writing a disk/CF sector are shown in Figure 3. At ①, the host provides the head, cylinder and sector numbers and the first word of data to be stored in the sector. The write control line is raised and then there is a delay as the disk/CF addresses the given sector. At ②, the first data word is written from the hDIn bus into the sector and the done signal is raised. At ③, the host lowers wr before the next rising clock edge. Then the host provides the next data word and raises the wr input at ④ to write it into the sector. The ATA controller will respond quickly by raising done at ⑤ after which the host lowers the wr input at ⑥. This procedure is repeated an additional 254 times to write the contents of the entire 256-word sector.

## ATA Controller Test Application

A simple memory tester application demonstrates the use of the ATA controller core. The memory tester performs the following functions:

- It initializes a pseudo-random number generator (RNG) with a known seed value.
- It writes a sequence of 256 random numbers into a single sector of the disk/CF.
- It re-initializes the RNG with the seed value.
- It reads back the contents of the disk/CF sector and compares it to the sequence from the RNG. Any mismatch indicates an error reading or writing the disk/CF.

The source files that describe this application are:

**common.vhd**: Some functions and definitions useful in many applications are provided in this file.

**memtest.vhd**: The memory tester state machine is described in this file.

**randgen.vhd**: This file contains the RNG used by the memory tester.

**atacntl.vhd**: This file describes the core state machine of the ATA controller and the low-level PIO interface.

**atatst300.vhd**: This is the top-level file that combines the generic memory tester and the ATA controller core to make the complete test application for the XSB-300E Board. You can set the operating frequency and the range of memory to test at the top of this file.

**xsbfpga.ucf**: The pin assignments for mapping the ATA controller test application to the XSB-300E Board are listed in this file.

**atatst300.npl**: This file tells WebPACK 6.1 how to combine the source files to create the ATA controller test application for the XSB-300E Board.

***Testing a Compact Flash Card***

1. Create a bitstream for the ATA controller test application targeted at the XSB-300E Board.
2. Place a shunt on the INT position of jumper JP8.
3. Place a shunt on jumper JP6.
4. Place a shunt on the RST# position of jumper JP10.
5. Insert a Compact Flash card into the CF1 socket.
6. Apply power to the XSB-300E Board.
7. Download the FPGA bitstream from step #1 using GXSLOAD.
8. Upon completion of the bitstream download, LED1 and LED2 will begin to glow as data is written to and read from the CF card.
9. Within a few seconds, LED1 should display a “0” if the test was successful. An “E” will appear if there was a mismatch between the data written to and read from the CF card.

***Testing an IDE Hard Disk***

1. Create a bitstream for the ATA controller test application targeted at the XSB-300E Board.
2. Place a shunt on the INT position of jumper JP8.
3. Connect an IDE hard disk to connector J3 on the XSB-300E Board. (Make sure the red wire of the IDE cable goes to pin 1 at the lower left-hand corner of J3.)
4. Apply power to the XSB-300E Board and the hard disk.
5. Download the FPGA bitstream from step #1 using GXSLOAD.
6. Upon completion of the bitstream download, the decimal point on LED1 will glow for up to twenty seconds as the hard disk is reset.
7. The decimal point will go off and LED1 and LED2 will begin to glow as data is written to and read from the hard disk.
8. Within a few seconds, LED1 should display a “0” if the test was successful. An “E” will appear if there was a mismatch between the data written to and read from the hard disk.

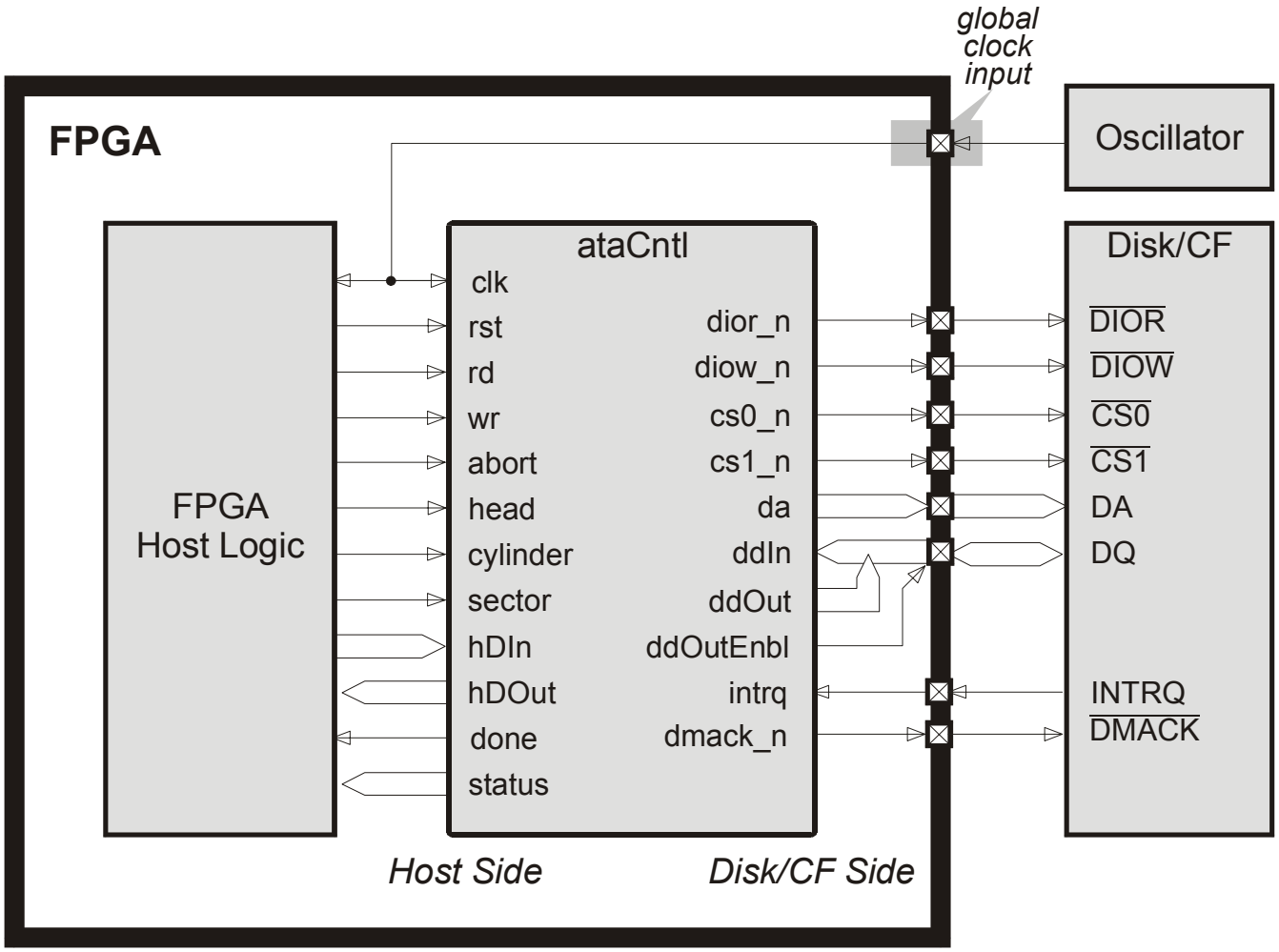


**Figure 1: ATA controller connections.**

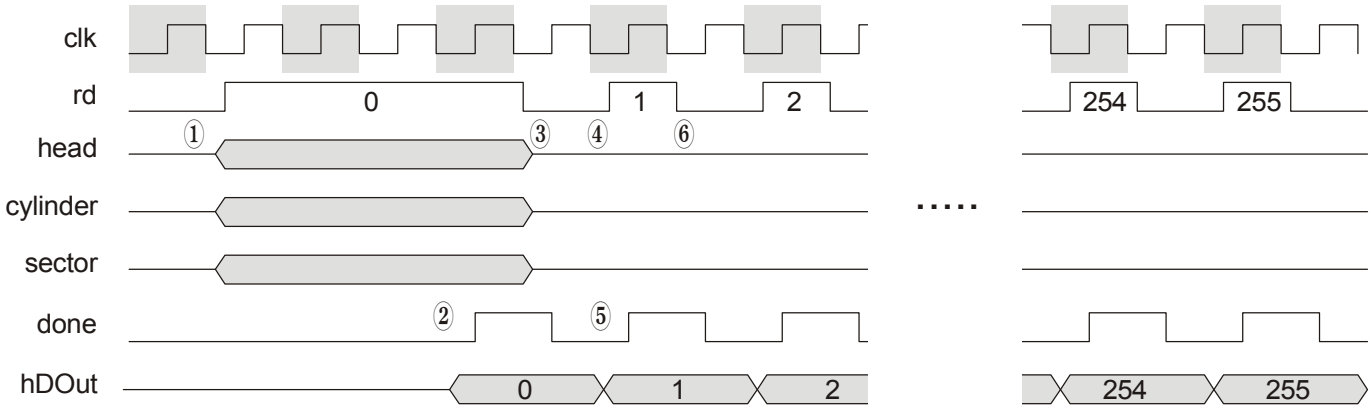


**Figure 2: ATA controller read sector waveforms.**

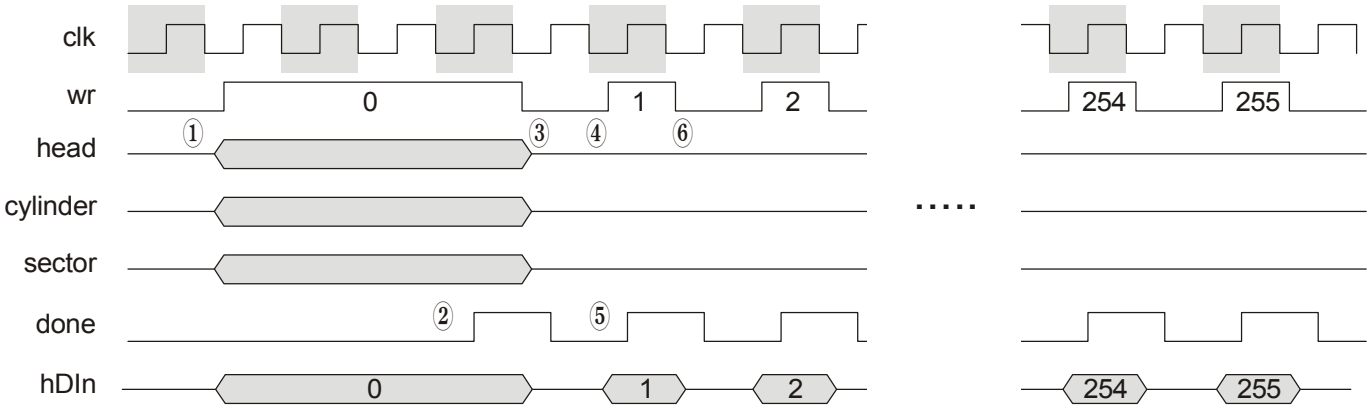


**Figure 3: ATA controller write sector waveforms.**