



## Release Notes

# **WITE32**

**Version 4.01**

**10/09/2008**

# CHAPTER 1

## INTRODUCTION

The release 4.01 of WITE32 incorporates new features and bug fixes introduced after the release version 4.00. This document uses the release notes for WITE32 version 4.00 as a baseline for comparison.

# CHAPTER 2

## NEW FEATURES INTRODUCED IN WITE32

### 2.1 Use of WITE32 Environment Variable for ATI Track Pitch Parameter

The *ATI* and *ATI MT* tests are modified to accept an environment variable set by other module as the *Track Pitch* value. Now for the *Track Pitch* parameter on the test configuration dialog box you can either enter a numeric value as usual, or a symbolic value started with '\$' symbol (for example, *\$ATIPitch*), see Figure 1.

The screenshot shows the 'ATI Test (Zone1:Setup1) - Wite32' configuration dialog box. The 'Track Pitch' field in the 'Write Options' section is highlighted with a blue dashed border and contains the symbolic value '\$ATIPitch'. Other fields include 'Runs' (1), 'Measurement Type' (TAA), 'PES Threshold' (0.1969 µIn), 'Succeeded Sectors' (80%), 'Max Retries Per Revolution' (6), 'Max Additional Revolutions' (400), 'Test Tracks' (Pattern: FF, Flux: 150 MFlux/s), 'Interference Tracks' (Pattern: LF, Flux: 100 MFlux/s, Revolutions to Write: 100), 'Measurement Repetitions' (1), 'Run Configuration' (Erase Before Test checked, Track Profile Plots: Draw TAA Plot checked, Plot Step: 0.6299 µIn), 'TAA Configuration' (TAA Iterations: 2, Filter: Overwrite), 'SpiSNR Configuration' (Spectrum From: 1 MHz, Spectrum To: 400 MHz, Spectrum Step: 0.1 MHz, Rejection Range: ±1 MHz), and 'BER Configuration' (Sampling Level: 6, Filter: F1). Buttons for Save, Reset, Default, Zone Edit, Close, and Run are visible on the right.

Figure 1: *ATI* test setup.

If you enter a numeric value, then both tests work exactly as in previous WITE32 versions.

If you enter a symbolic value, the tests consider it as a name of a WITE32 environment variable (*ATIPitch* in this example). When the tests are executed they evaluate this variable and use the value of the variable as the *Track Pitch* parameter. The tests treat this value in  $\mu\text{Inch}$ , if the unit system in the product is British, and treat it in nm, if the unit system is metric.

The value of the WITE32 environment variable must be defined before the tests are executed; otherwise, a run-time error will occur. To define a WITE32 environment variable, you can either develop a custom WDK test or use the WDK script.

If you develop a custom WDK test to set the variable, this test must be executed at least once before *ATI / ATIMT* tests. Inside your custom WDK test, you use the *WUTLSetEnv()* WDK function to set a WITE32 environment variable. For example:

```
Call WUTLSetEnv("ATIPitch", "12.345")
```

The above statement will set the WITE32 environment variable, *ATIPitch*, to 12.345.

If you use the WDK script to set the environment variable, you can call the same *WUTLSetEnv()* function from the scripting environment.

## CHAPTER 3 FIXED BUGS

The following bug was discovered in WITE32 version 4.00 or earlier, and fixed in WITE32 version 4.01. The description of a bug explains the bug behavior as it appears in WITE32 version 4.00.

1. The head amplifier chip physical damage may occur due to continuously set write gate signal. The continuously set write gate signal switches the head amplifier into continuous write mode, which in turn may burn the head amplifier chip. It can occur if the currently selected system pattern is of PRML type and one of the following conditions are met:
  - When a specific PRML chip adapter is selected in the Product Configuration dialog box, but the chip adapter board is either not installed or is incompatible with the selection.
  - When the loading of the chip adapter FPGA fails or is not performed at all due to incorrect license check.
  - When AutoDetect option is selected in the Product Configuration dialog box, but no chip adapter board is installed.