

**SiliconAid Solutions SAJE JTD™ JTAG INTERACTIVE DEBUGGER**

**JTD™ Product Overview**

- Drives JTAG hardware 5 pin interface for chip debug
- Multiple Hardware support - USB, Ethernet, and more
- Drives Evaluation boards, Apps boards, ATE and more
- Compares expected values for TDO
- Supports run till FAIL, STEP, and much more
- Leverages Design Data passed from JTV if available
- Supports External SVF Patterns
- Both Windows & Linux platforms are supported

**JTD™ Product Description**

SAJE JTD has been developed to be an interactive JTAG debugger for chip debug and troubleshooting. (Figure 1) Easy and intuitive to use, JTD can read in patterns from multiple sources and drive the corresponding data out of the USB port. JTD can capture actual data from the device and compare it to expected data.

Developed to have the same basic look and feel as a simple ATE tester to control and observe data. (Figure 2) JTD has many of the same features: **Run, Step, Stop on Vector, Stop on Fail, and more...**

Special modes have been built into JTD to allow leveraging of design data and information in the BSDI file. This data is automatically displayed with intuitive graphical windows. These windows are synchronized to the active vector and change dynamically as you step through the vectors.

JTD also tracks the JTAG state machine dynamically and displays it status (Figure 3).

SAJE JTD can create custom test sequences specified by the user. Combining or rearranging individual tests into a custom sequence to be stored or ran on hardware.

**CUSTOM PATTERN SEQUENCE**

JTD can display failing data from TDO and link it to the corresponding internal register that captured the fail. Thus eliminating the need for the engineer to trace backwards to find the source of the error. (Figure 4)

**CAPTURE MODE**

SAJE JTD provides an automated capture mode utilizing the power of JTAG and scan modes together. This unique mode allows the user to run a test and specify internal registers to automatically generate the waveforms of those internal registers. Giving the user the ultimate logic analyzer function displayed in an easy to read waveform. (Figure 5)

Figure 1 – JTD main window



Figure 2 – Debugger control

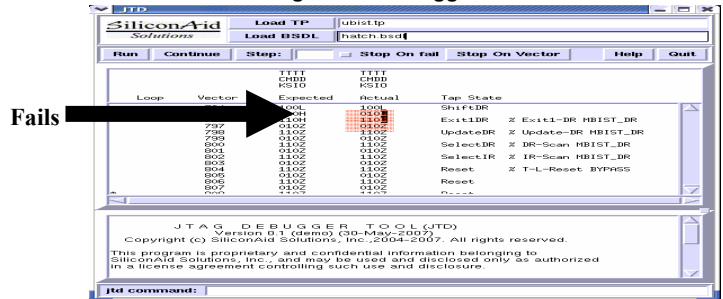


Figure 3 – Register Viewer

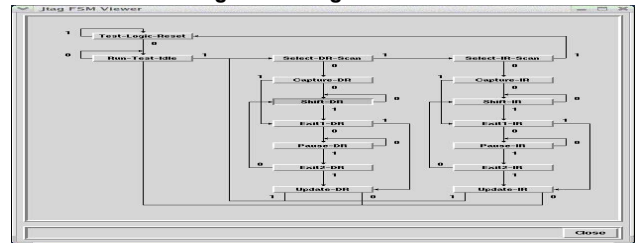


Figure 4 – Register Viewer

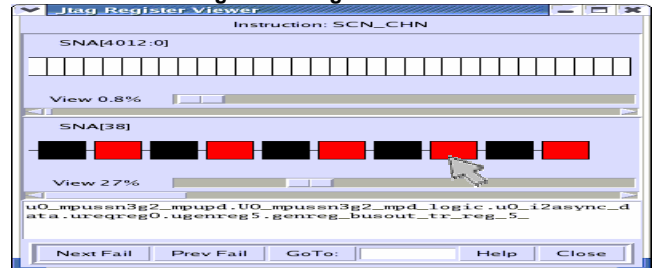


Figure 5 – Waveform Viewer

