Using the Silicon Explorer For System-Level Debug

Problems found late in the design cycle can be time-consuming to debug and correct and can significantly impact product schedules. This problem is exacerbated by the ever-increasing complexity of designs and densities of programmable devices. The purpose of this application note is to describe Actel's Silicon Explorer feature and how it can be used to minimize schedule impacts resulting from functional or timing problems discovered late in the design process.

The Design Process

Much of the design process is spent in functional verification (simulation) and timing verification. Functional verification typically consists of modeling external devices that provide stimulus to the design and ensuring that the design responds properly. Timing verification can be performed either by backannotation or static timing tools like Actel's DirectTime Analyzer.

Simulation is a typical GIGO (garbage in, garbage out) function, because accurate verification of the design is completely dependent on the accuracy and completeness of input stimulus and expected response. Lack of completeness can be an issue if certain cases are not tested. Test cases are sometimes overlooked or intentionally left out because of complexity or to limit tests to corner conditions.

Timing verification presents similar challenges because correct function is dependent on other devices meeting timing expectations included in the test cases. Timing verification is typically limited to corner conditions to ensure manageability of the design time, which by definition ignores potential timing problems at non-corner conditions.

Where the Rubber Meets the Road

No matter how much effort is put into functional and timing verification, there is no substitute for powering up the system and seeing it operational. So, here you are late in the design cycle. You get your prototype boards built, you apply power to the system, and nothing happens (at least no smoke). You pull out your logic analyzers and emulators and start trying to debug the problem. You first verify connectivity and make sure clocks are wiggling. You keep working until you discover that the outputs of your FPGA are not functioning.

Now you have several choices:

- Spend a few days verifying that the inputs to the FPGA correspond to your simulations. If not, then go modify simulation inputs, and see if you can replicate and then fix the problem.
- Guess what the problem is, based on the observable failure mechanism external to the device, and then attempt to correct it.
- Place and route the design again, and hope the problem goes away, or perhaps bring out some test signals if you have unused pins.
- Or, if Actel is your FPGA solution, you can use the Silicon Explorer diagnostics and have your problem exactly identified and corrected in a few hours.

The Silicon Explorer Diagnostics Tool

Actel's Silicon Explorer provides the ability to probe any internal node during normal device operation. The internal node is selected by scanning a serial address from your computer into the device on the pins SDI and DCLK. Once the addressing is complete, a path from the internal node to either the PRA or PRB I/O pin is created. A logic analyzer connected to these pins can then be used to observe the internal operation of the device eliminating the guess work of system debug, or the signals can be observed using the PC logic analyzer.

To select an internal node to observe, simply select the net name or pin name corresponding to the node. The tools will then calculate the required serial stream and communicate this information to the device. Up to two nodes can be addressed simultaneously and moved around as quickly as alternative nodes can be selected. So, starting from a failing output, you can quickly trace back through the device to determine the origin of the failure. Once the problem is found (whether functional or timing), the design can be quickly modified, allowing you to rapidly achieve a working system.

Examples

Silicon Explorer can be used when the simulation stimulus is incorrect or incomplete. Incorrect stimulus (accidental or intentional) can be quickly discovered by using Silicon Explorer to locate the earliest failing section of logic. The system stimulus for this small section can then be compared with the simulation stimulus to determine differences and corrective actions.

Large counters present another example where simulation may be inadequate. For instance, a pulse is required every second and must be generated from a 20 MHz clock. This means you would need a 25-bit counter and 20 million simulation cycles to verify the counter, which could take a while. The typical approach is to set the counter near the decode/reset point, run the simulation from there, and assume that the counter will work otherwise. If for some reason there is a failure in the counter circuitry at some unsimulated point, Silicon Explorer can be used to locate the problem quickly by providing the ability to observe all counter states, next states, and decode values.

Security

If security is a concern in the final product, additional fuses can be programmed to disable the Silicon Explorer function.

Requirements

The hardware required to support the Silicon Explorer diagnostic tool includes the Silicon Explorer pod, which connects the device under test to the PC through the serial port. In addition, the MODE pin on the device must be connected to ground through a 10K resistor. The software to drive the Silicon Explorer is built into the APSW software, included with all Designer Series Development Systems.

Limitations

The Silicon Explorer diagnostic capability has a maximum observable frequency that is family dependent, as shown in Table 1. In addition, the signals PRA, PRB, SDI, and DCLK should be available (not used as I/Os) to fully take advantage of the Silicon Explorer capability.

Table 1 • *Maximum Observable Frequencies*

Device Series	Fmax
ACT 1 (1.0 µm)	25 MHz
Integrator Series	45 MHz
Accelerator Series	80 MHz

Conclusion

Increasing complexity of designs and decreasing cycle times create a significant challenge to the designer to produce a working system. Strategies to meet this challenge include simulation and timing verification tools. When these tools fall short because of a lack of accuracy or completeness, good system debug tools are required, including emulators and logic analyzers. Finally, to bare the internal operation of a device to scrutiny, Actel has developed the Silicon Explorer function to remove the guesswork from device debug. The result is improved productivity and products getting to market faster.