Univers® COVER is a complete SW/HW co-verification tool for multi-processor SOC architectures, offering extensive debug capabilities, fast simulation and full speed target debugging support. It includes an RTL simulator as well as a multi-core IDE with most fascinating debug features and a configurable user-friendly graphical user interface. Adveda also provides fast cycle-accurate processor Instruction Set Simulators (ISS), which enable these features. Univers COVER is the Integrated System Co-verification tool of Adveda’s UNIfied VERification Solutions.

Univers COVER offers one central cockpit from which the user can oversee his complete system-on-chip. It verifies the software running on multiple processors as well as the hardware blocks surrounding these processors. Univers COVER is the ideal solution to have the System Engineer, Software Engineer and the Hardware Engineer work together with one and the same tool.

Here is a short summary of the main features:

- Fast cycle-accurate Instruction Set Simulators
- Combines multiple different ISS’s
- Complete Integrated Development Environment
- Automated flow for multi-core SOC projects
- Ultra Fast Verilog/VHDL simulation
- Extensive and unique debug features
- Graphical static and dynamic profiling
- Open debug-API for integration
- Multi target (ISS, RTL, HW)
- Automatic embedded OS support
Meet the family

Univers COVER is the top-line product of Adveda’s Univers family and combines Univers RTSIM a standalone RTL simulator and Univers ISSIM, a multi-core software debugger, into one unique SW/HW co-verification tool, which can run a complete system-level cycle-accurate simulation of both hardware and software at an unsurpassed speed.

Univers COVER provides a multi-window interface with which developers can activate desired functionality and select system visibility as the debugging requirements evolve. It also integrates the development tools (C-compiler, assembler, linker) for each processor and manages the flow between these tools. The flow for hardware compilation is identical to the software approach and many debugging features of both the SW and HW domain can be interwoven. All usual run-control functions are provided through menus and icon-buttons within the context of different windows. All settings of your project are saved in a XML-formatted project file.

The Project Window instantiates the project and controls the overall debug process. Every processor core and RTL peripheral in the project is instantiated in this window. Each processor or peripheral may have its own list of features and relevant debugging windows.

The project window also contains a logging field as well as a command prompt. Furthermore, clocks, IO, DMA, breakpoints as well as inter-processing communication can be setup via this window.

Each processor has an Application Window, which contains the software source codes of the project for that processor. The object code can be built with all related files or can be individually compiled and assembled, under menu or button control.

In the Application Window, the source code can be viewed, edited and (re-) compiled. The source code is context-sensitively colored for ease of reading. For easy debugging, the values of variables pop-up when the mouse is moved over the associated source code text. The Application Window also allows a walk through a function’s backtrace, while simultaneously updating the Variables Windows, Stack Windows and Application Window itself, thus facilitating easy tracing of function calls. Obviously, breakpoints in your source-code text can be placed as well.
SOC verification at unsurpassed speed

Every HDL design has a Model Window, which contains the RTL source codes of such a peripheral. In the Model Window, the RTL code can be viewed, edited and (re-) compiled. The RTL code is context-sensitively colored for ease of reading. For easy debugging, the values of signals pop-up when the mouse is moved over the associated source code text. Breakpoints in your RTL code can be placed as well.

Every memory space may be made visible in multiple windows. In such a window one may set all types of breakpoints at every individual location or simultaneously at a range of locations. In every Memory Window one can define how a memory is instantiated and/or shared with other processors or peripheral blocks.

Every processor has its own set of registers, which are displayed in a separate processor Register Window. It is also possible to display the actual registers of the RTL hardware peripherals in the same or another register window. Besides just viewing the content of registers, it is possible to overwrite these or set various types of breakpoints for each register. The register contents are color-coded indicating the type of usage of the registers since last execution.

Univers COVER automatically detects stack operations as well as different thread actions and will store both into the Stack Window. This allows Univers COVER to support any OS on any type of processor. This is one of the unique features of this integrated development environment.

The Profile Window collects all data at run-time and it may be viewed at each moment in time. The profiler displays execution cycles, stall cycles and types of stalls for each function in a very convenient way, highlighting the options for optimizing the application programs. The profiler supports graphical static profiling as well as graphical dynamic profiling.
Every processor may have its own Disassembly Window containing a lot of information about the execution of the program. It doesn’t only show the disassembled code, but also provides information how many times instructions are executed, how many stalls are introduced by each instruction and which effect caused the stall. This provides all basic information, which can be used to improve the overall performance of your application.

Within the Waveform Window every signal of the RTL code can be viewed. Also any value within the ISS of each processor or each peripheral can be traced in the same or another Waveform Window. This enables the display of the history of selected signals and may give a very good understanding of detailed cycle aspects. It allows to debug and fine-tune the application code together with RTL signals.

All variables are displayed in the Variables Window. Variables may be selected per procedure or from a user-defined list. The window shows the value of a variable as well as his absolute address in one of the associated memory spaces. For RTL signals there is a similar Signal Window.

Univers COVER is a complete solution for hardware and software co-design and verification. When used together with the fast Instruction Set Simulators from Adveda, all debug features can be used. Other Instruction Set Simulators may be integrated too, but might have less debug capabilities. Univers COVER also supports interfaces to real hardware implementations of the processors. Some debug features may not be available in such mode since not all data, available from the ISS models, can be obtained from real hardware implementations. The amount of debug features will vary per processor implementation.