WIND RIVER

Wind River Workbench On-Chip Debugging 3.3.2

Wind River offers the industry's leading development environment for hardware-assisted debugging. Wind River Workbench On-Chip Debugging takes you from early hardware bring-up to test and manufacturing with a centralized development solution. The Wind River Workbench On-Chip Debugging development environment powers the Wind River ICE 2 and Wind River Probe JTAG debug units.

The introduction of 32-bit and 64-bit processor technologies has created new challenges that require an innovative approach to JTAG debug and analysis. Wind River offers an Eclipse-based development environment for hardware-assisted debugging that encompasses all phases of your device development life cycle. The advanced hardware diagnostics and patent-pending multi-core debugging capabilities of Wind River Workbench On-Chip Debugging help you unlock the power of today's advanced microprocessor technology.

Wind River ICE 2
High-performance JTAG debug unit for the leading ARM, Intel Architecture, MIPS, and PowerPC uniprocessor and multi-core devices

Wind River Probe
Entry-level, portable JTAG probe for debugging ARM, Intel Architecture, MIPS, and PowerPC devices

Wind River Workbench On-Chip Debugging
Extension of the Wind River Workbench development environment, with JTAG on-chip debugging capabilities that power Wind River ICE 2 and Wind River Probe; based on the industry-standard Eclipse platform
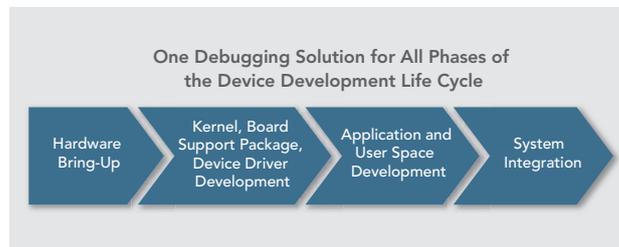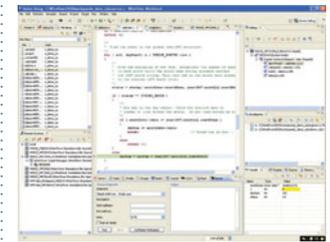
One Debugging Solution for All Phases of the Device Development Life Cycle

Hardware Bring-Up → Kernel, Board Support Package, Device Driver Development → Application and User Space Development → System Integration

Figure 1: Wind River's on-chip debugging solution

Wind River Workbench On-Chip Debugging can be easily extended through in-house, third-party, open source, or commercially available Eclipse plug-ins. For example, you can integrate software version control and change management solutions such as CVS and ClearCase. You can also integrate UML-based software visualization and code generation tools, enhanced static analysis solutions, and simulation/virtualization software.

Key Features

- Debug the most complex 32-bit and 64-bit multi-core processors.
- Simplify workflow of the edit-compile-debug cycle with the centralized project-oriented environment and advanced editing and debugging capabilities.
- Streamline connection to multiple targets with advanced connection and configuration management.
- Program flash memory with a wide variety of provided algorithms.
- Support a wide range of processor architectures including ARM, ColdFire, Intel, MIPS, and PowerPC.
- Extend easily through the Eclipse-based development environment to support third-party and open source tool integration.
- Operate via command-line interface options and extensive scripting capabilities.
- Debug complex, intermittent code defects with internal and external trace buffer support (for supported processors).
- Take advantage of target OS awareness for the latest versions of Wind River VxWorks, Wind River Linux, and open source Linux kernels.

Table of Contents

Wind River Workbench On-Chip Debugging enables you to take advantage of the growing Eclipse ecosystem to improve the collaboration between hardware, firmware, and software developers, ensuring a smooth handoff in the debug process.

Figure 2: Wind River Workbench On-Chip Debugging
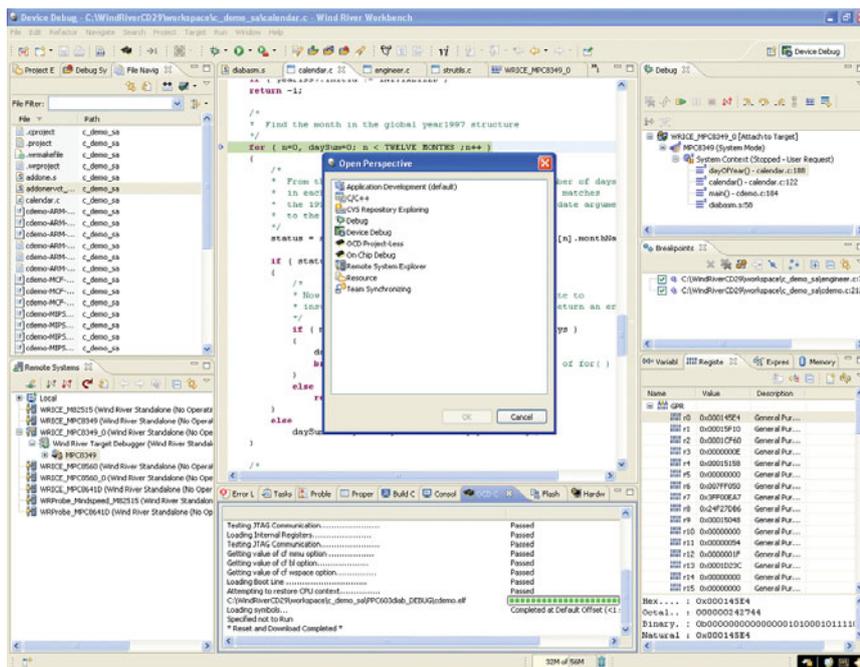
Figure 3: Specify perspectives relevant to a specified task

Centralized, Project-Oriented Environment

Workbench On-Chip Debugging provides a centralized, standards-based development environment that automates the debugging and analysis process. Its project-oriented methodology simplifies the handoff between the different phases of development, from initial hardware bring-up to firmware design, OS implementations, and platform and application development.

The Workbench development environment efficiently manages hardware and software development projects. Its flexible, configurable project interface enables you to customize project data based on type of project and development activity. Wizards and utilities walk you through the creation of new projects to get you up and running quickly.

From a single console, you can see information about multiple project files, multiple targets, and the software running on those targets. Because different stages of development require different debug views and capabilities, you can specify a particular developer perspective that provides only the information relevant to a specified task.

The on-chip debugging perspective delivers the starting point for JTAG debugging, including the views required for connection to Wind River JTAG debug units, as well as the specific tasks associated with JTAG-based debugging.
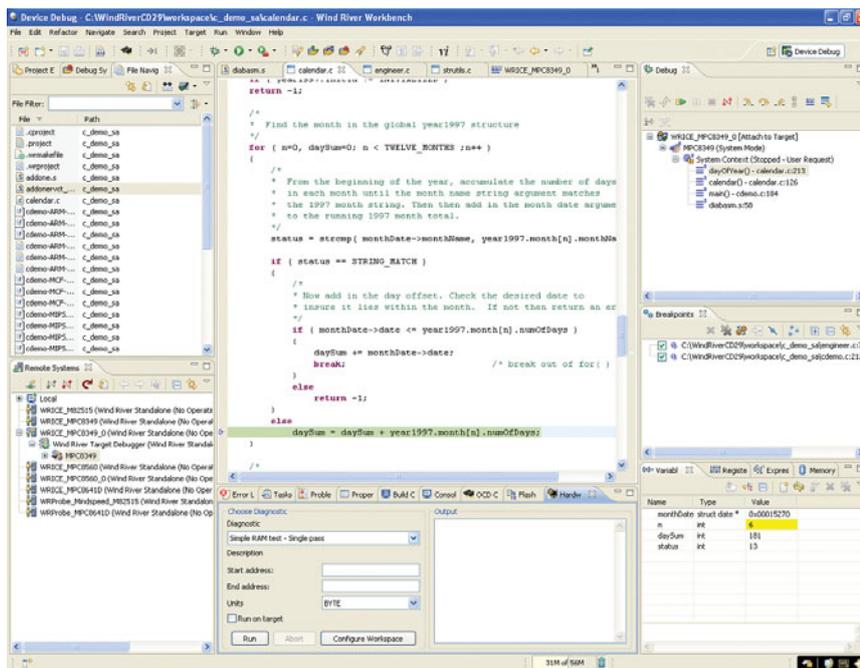
Perspective in Eclipse

Perspective is a specific term in the Eclipse platform that refers to the visible actions and views within a window. It includes an input attribute, which defines the resources in a workspace, and a type attribute, which defines the actions and views in the user interface.
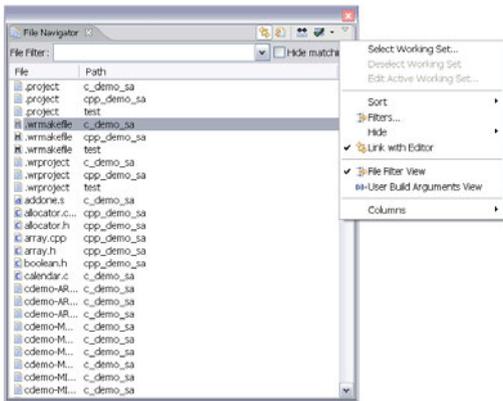
Figure 4: Workbench file navigator allows you to quickly find symbols and functions

Project Navigation Capabilities

Advanced project navigation capabilities reduce the complexity of managing multiple projects. You can organize and manage all the software components in a device software development project. Moreover, you can create and share projects with customized debug and launch settings among other development team members, reducing setup and configuration time and improving collaboration. When combining software debugging with on-chip debugging, you can tighten the project handoff between hardware and software developers.

Integration with CVS enables you to identify when other developers are editing the same code in order to prevent editing conflicts. The Workbench file navigator feature also allows you to quickly search and find symbols and functions within all files loaded in the project workspace, shortening development time.

Comprehensive Build System

The Workbench On-Chip Debugging build system provides the tools, options, and parameters for device software build management, enabling you to define everything from global build parameters to fine-grain control of an individual file. By using a single build environment, you can reduce the time associated with managing multiple environments. For targets utilizing multi-core processors or multi-OS configurations, the build system includes intuitive wizards, settings, and options so you can become productive quickly. With build capabilities built directly into the software, you

have a centralized solution for the entire edit, compile, and debug process.

Developer Information Center

Workbench On-Chip Debugging contains videos and screen-casts to showcase Workbench fundamental operations including links to online help topics. Developers can review basic and advanced topics on how to create projects, debug code, analyze code, and use Workbench to collaborate efficiently in a team setting.

Advanced Editing Capabilities

The editor provides state-of-the art editing, including vi emulation. Performance-enhancing features, such as code completion, parameter hinting, and syntax highlighting of source files, improve edit-compile-debug cycle time and reduce errors. The editor is tightly integrated with the project system, build system, Source Code Analyzer, and debugger in Workbench, enabling you to move back and forth easily between various source-code debugging tasks.

JTAG Editor

In addition to software editing capabilities, Workbench On-Chip Debugging offers a JTAG editor to create and modify board configuration files. The editor allows you to connect a Wind River JTAG emulator to a board with multiple devices on its scan chain. The editor then provides a graphical view of the layout of devices on the scan chain, expediting the update of board configuration files.

When you only have one or two devices in a scan chain, it is easy to set up the connections and configurations. As devices are added, complexity increases—and so does the chance for error. A graphical view enables you to easily see and configure which devices are in the chain, so you know how to configure the emulator to support the correct number of devices. The JTAG emulator will also need to know how to address the various devices on the scan chain, so it can access

specific devices for a debug task while the rest of the devices are set in bypass mode. The editor saves this information and downloads it to the emulator.

Connection and Configuration Management

Workbench On-Chip Debugging centralizes all target configuration and connection management in a single graphical view, eliminating the need to toggle between multiple GUIs or manually configure connections via CLI. These connections can be cores, processors, processes, or simulators. You can manage multiple connections simultaneously or group multiple cores as a single connection.

Eclipse Launch Capability

The Quick Target Launch Utility in Workbench On-Chip Debugging leverages the powerful Eclipse launch capability, enabling device software developers to easily define cross-target launches. Developers may also select from a list of predefined launches, either taking control of an already running target or starting a target from reset, which results in the user's quick movement from selecting target launch to debugging complex hardware and software problems.

Launches may be used to establish a target connection, download an image to the target, load symbol information into the debug engine, or run scripts or any combination of these tasks. In addition, users may create batch launches, which group together individual launches and enable Workbench On-Chip Debugging to establish connections or download software to multiple cores or processors simultaneously.

On-Chip Debugging Command Shell

If you prefer using CLI, the On-Chip Debugging Command Shell provides CLI access to the target via a Wind River emulator. The Command Shell and associated logging capabilities allow you to make target and emulator configuration adjustments. You can also write low-level scripts for target initialization and download sequences to automate commonly occurring tasks.

Figure 5: Graphical display allows visualization of all CF options associated with the Wind River emulator for a specified target

The Command Shell is used to load target register files into the emulator then execute low-level commands through the emulators. Sessions in the Command Shell can be recorded and replayed as scripts. There are two command log options: input logging and full logging. Input logging only records input commands while the full logging option records input commands and their associated results. These files can be saved and played back later.

Console for Target Status Reporting

Workbench On-Chip Debugging includes a console specifically for JTAG connections, providing data on target connection status and the events executing on the target, verifying that a given task executes as instructed.

Graphical Display of All Configuration File Options

To expedite the configuration process, Workbench On-Chip Debugging provides a target register configuration file (CF) for the majority of supported processors. You can customize these configuration files using the Workbench On-Chip Debugging CF Options view.

With the graphical display, you can visualize all the CF options associated with the Wind River emulator for a specified target, including command name, current setting, parameters, and description data.

Expanded Register View Capability

Get your target up and running quickly with ability to access and view bit-level detail for CPU and peripheral registers. With a single click you can view and configure peripheral register bit-fields to meaningful values. This eliminates the time-consuming task of sorting through processor documentation. In addition, you can create your own peripheral register groups and modify preconfig-ured register files to more effectively manage custom configurations.

Enhanced Capabilities for Intel Architecture

Advanced features specific to Intel architecture CPUs allow you to quickly assess the state of your Intel platform and efficiently debug it. Visibility is provided into the IA system structure, CPU, and peripheral registers. Specific features include the following:

• IA descriptor tables (GDT, IDT, LDT)
• IA paging tables
• Physical-to-virtual memory address translation utility

• Interactive bit-field editor with bit-level detail and inline documentation for CPU and peripheral registers
• Intel MMX technology registers represented in floating-point format
• Ability to set breakpoints on I/O port 80
• Auto-enumeration of devices on the PCI bus
• Ability to program SPI flash devices
• Capabilities to debug UEFI-compliant BIOS and boot loaders

Binary Upload and Compare Utility

The Binary Upload and Compare Utility enables developers to quickly extract information from any area of memory on the target system, assuaging worries about overwriting the boot ROM or boot loader shipped with the processor vendor's reference board. This capability is particu-larly useful when trying to save/store boot ROMs or boot loaders that may be shipped with a target for safekeeping on the host PC.

The Binary Upload and Compare Utility enables developers to graphically select an area of memory on their devices, including sectors of flash, and upload the images into a file on their host PCs. Once the image is stored on the host PC, it can be used by the Compare Utility to identify and resolve file corruption issues on the target system. The Compare Utility compares images on the target with the content of binary files stored on the host PC. Any differences between these files are recorded, and the information is displayed in the Work-bench On-Chip Debugging Editor view.

Analysis and Diagnostics

Advanced Hardware Diagnostics

At the hardware level, Workbench On-Chip Debugging provides compre-hensive hardware diagnostic capabilities that eliminate the need for certain

standalone hardware diagnostics tools, reducing capital and training costs. By integrating preconfigured test routines and scope loop tests into a single interface, you do not have to spend valued development time writing your own test tools for data and addressing bus diagnostics.

Wind River provides robust and integrated software that quickly isolates and resolves hardware layout and bus issues. You are guided by a set of wizards to configure and run the various diagnostic utilities:

• Address and data bus tests to verify address and data bus performance and quickly identify and resolve issues
• Cyclic redundancy check (CRC) calculations for a defined section of memory
• RAM tests ranging from simple to full: writing a consecutive pattern of three values into a defined area of memory then reading it back for errors, or executing in a single pass or continuous test pattern
• Scope loop tests that put a known pattern on the address and/or data bus for monitoring by an external data scope

Source Code Analyzer

The Source Code Analyzer simplifies the process of documenting the code structure of a file system to support effective integration of large project file systems. This is especially valuable when multiple developers are writing code; code is reused among projects; or newly developed code must be integrated with a legacy code base. You can quickly and completely understand the code written by someone else and integrate it effectively into the existing project. In addition, you can instantaneously see the impact of a proposed change in the existing code, improving overall development productivity and reducing errors.

Statistical Code Profiling

Built-in performance analysis and code coverage profile software identify system bottlenecks to optimize software execution on a target processor. Unlike a typical profiling solution, no configuration or instrumentation is required. It is easy to select a function from the statistical code

profiler then see where the function resides in the Source Editor view.

Profiling is as simple as identifying a section of code. The Workbench On-Chip Debugging software does the rest, displaying the results in a chart, table, or histogram that includes the following:

• Function name
• Full path to the function
• Start and end addresses for the function
• Percentage of time within the function for all functions to be executed on the target

Cache Memory Analysis

Tracking cache coherency issues with main memory can be a challenge for a single-core device application. The complexity of multi-core implementations makes it almost impossible without sophisticated cache memory analysis. The cache analysis capabilities in Workbench On-Chip Debugging monitor execution on one or more targets and identify differences between data stored in memory versus data stored in cache. You can quickly identify the source code and function for each instruction in cache and easily toggle to its location in the Editor/Source Code view or symbol browser.

Comprehensive JTAG Debugging Engine

Debugging is the most time-consuming and costly phase of any development cycle. Wind River provides a single console for debugging multiple projects across multiple targets. With these debug collaboration capabilities, different developers can easily share information, dramatically improving debug cycle time.

The advanced debug engine supports multiple debug connection types, including JTAG tools, agents, and simulators, providing maximum flexibility in debugging both hardware and software from the early hardware diagnostic stages to application development. You can debug from any host OS to any target OS. Multi-core debugging is easier because you can set cross-correlated breakpoints and track information across multiple targets.
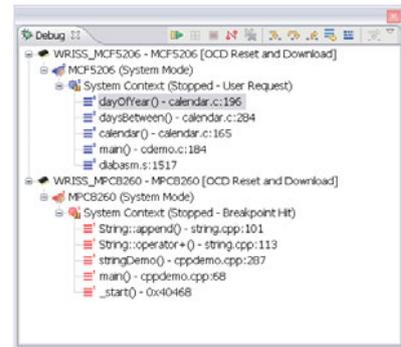

Figure 6: Debug view monitors, controls, and manipulates active tasks

Debug visualization shows at a glance the processors being debugged and the active debug tasks through color-coded and numbered debug sessions. It provides the following information:

• **Standard run-control capabilities:** Start/Resume, Stop/Suspend, Terminate, Step-In, Step-Over, Step-Return, Source Mode, Function Mode, and Assembly Mode debugging
• **Data on the target or process attached to the target:** Stack frame, threads, processes, and tasks
• **System information:** Memory views, Register views, Watch view, local and global variables
• **Multi-core Debug view of the stack frame:** Every process or target running under debugger control displayed in the Debug view, color-coded, and numbered

Common Scripting Framework

Workbench On-Chip Debugging comes with a Host Shell scripting framework to control all debugger activities. It supports standard scripting languages, such as GDB command syntax for low-level debug commands and Tcl or C interpreters for high-level flow control. The Host Shell is supported on Linux, Solaris, and Windows hosts.

Breakpoint Options

Breakpoints are a valuable tool used to stop the target or process running on the system when a specific user-defined event occurs. Workbench On-Chip Debugging supports hardware and software breakpoints configured as either data or expression breakpoints. A graphical view of all breakpoints on a
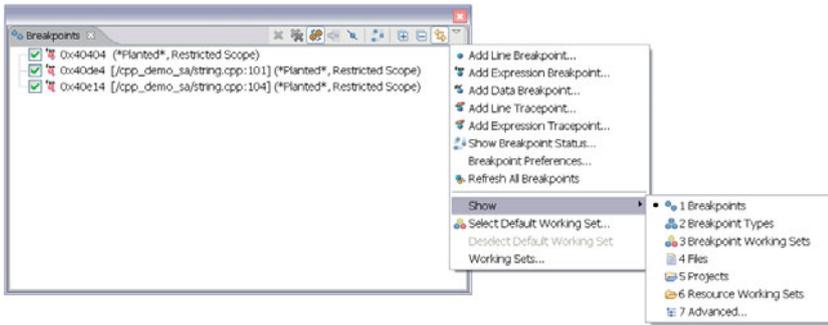
Figure 7: Graphical view of all breakpoints on a target improves management of multiple breakpoints

target provides more effective management of multiple breakpoints. You can set breakpoints from the Breakpoint, Source/Edit, Register, and Memory views.

Specific breakpoint features include the following:

- Hardware breakpoints
- Software breakpoints
- Expression, line, or data breakpoints
- Enablement and disablement of any event in the system
- Save to a file and redeploy among multiple projects
- Multi-context awareness, allowing breakpoints to stop the specific core or processor in context with an event
- Ability to stop the entire system when the event the breakpoint is associated with occurs
- Range of icons to manage breakpoint configuration and settings
- Host Shell script execution upon hitting a breakpoint

Flash Programming

Workbench On-Chip Debugging simplifies the configuration of flash memory, enabling you to store images into flash memory on the target board faster than the traditional manual CLI process allows. The flash programming utility supplies common algorithms and diagnostics to pinpoint flash programming issues, enabling you to do the following:

- Configure the flash address and RAM workspace, supporting flexible memory configuration and user-defined allocation of flash programming algorithms into RAM
- Choose files for download; you can select and manage the files to be flashed onto the target from a single console

- Execute erase and program operations by selecting one or more sectors
- Verify the resulting data residing in flash with the file to be flashed
- Program NOR, NAND, and SPI Flash devices

Support for new or unusual flash devices can be easily added by utilizing the provided sample source code and build instructions.

Trace Support

Workbench On-Chip Debugging provides support for internal and external trace buffer features for select processors. You can quickly spot challenging intermittent defects that take a long time to analyze with traditional debug methods. Wind River ICE 2 can be extended with the optional Wind River Trace 2 external trace buffer module.

You can view the events that occurred in the execution of the trace, such as the start of the trace and end of the trace, along with a description of the event. You can then select an event from the trace event field to locate the specific event occurrence in the trace buffer. Specific trace fields include the following:

- **Event occurrences (unlabeled):** Type of trace event
- **Address:** Address or line number of trace event
- **Absolute time:** Elapsed time since the beginning of the trace
- **Delta time:** Change in absolute time since the last trace entry
- **Instruction (unlabeled):** Executed instructions that can be configured to display code at the function, source, or disassembly stage

The flexible trace configuration options enable you to configure and set up the display of trace data, clear the trace buffer, reset the Trace view, save the trace to file, set trace rules, and configure trace filtering.

Multi-core Technology

Multi-core refers either to a single chip containing multiple core or threads capable of executing code, or multiple processors on a single target board. In addition, in system-level design, you may have multiple processors across multiple boards. The emergence of these multi-core processors is creating new debug challenges, requiring innovative technology that leverages the open environment of the Eclipse framework.

Multi-core Debugging Challenges

Multi-core offers more functionality and higher performance, but it also poses new debug challenges for hardware and software developers:

- How to effectively visualize and manage the edit-compile-and debug cycle across multiple cores
- How to optimize the JTAG interface with many SoCs, leveraging a single JTAG interface to save on costs
- How to manage real-time performance requirements for multi-core debugging
- How to support multiple processors from different vendors
- How to support the debug of multiple operating systems across different cores
- How to perform system-level debugging of hypervisor-enabled systems and virtual boards

Value of Eclipse in Multi-core Debugging

A common debug framework with a strong ecosystem is the foundation of a multi-core debug strategy. While some take a proprietary approach in integrated development environments, Wind River leverages the Eclipse framework to reduce the complexity of the edit-compile-debug process.

Workbench On-Chip Debugging offers a single project-oriented framework for debugging the most complex multi-core scenarios. It allows you to view all your multi-core projects from a single interface, whether the cores are on a single board or on multiple boards in an integrated system.

Target Connection Manager

An easy-to-use target connection manager simplifies the attachment to multiple cores and supports multiple connection methods, including Wind River JTAG debug units, Wind River's Transparent Mode Driver (TMD), a software agent, or a simulator. The Wind River solution connects to up to 128 cores in a system and debugs up to 16 cores or 32 threads simultaneously through a Wind River emulator JTAG connection, solving the problem of managing multiple cores through a single JTAG interface. In addition, the optional Wind River Connect 2 extension module for Wind River ICE 2 connects up to four different scan chains simultaneously to provide the scalability required in system-level multi-core development.

The target connection manager is based on the Eclipse Remote System Explorer (RSE) framework, providing you with remote access to target file systems through a number of network-based protocols
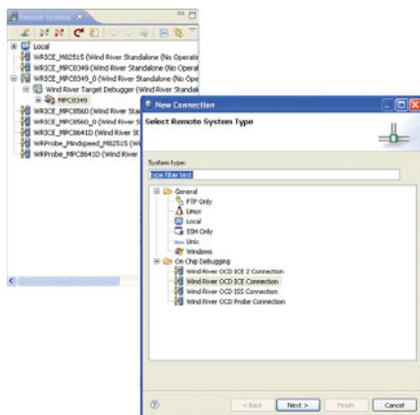

Figure 8: Target connection manager simplifies attachment to multiple cores and supports multiple connection methods

Advanced Debugging Features

When you're debugging multiple cores, managing and tracking hardware and software breakpoints becomes even more critical. Workbench On-Chip Debugging synchronizes the run-control of a system to start and stop the entire system. The Wind River ICE 2 JTAG debug unit leverages debug capabilities of the multi-core processor and manages multiple cores with a single JTAG connection. You can set cross-correlated breakpoints and track sessions with multiple targets. The debug sessions are color-coded and numbered, so you can quickly spot an issue on an impacted core.

Workbench On-Chip Debugging enables system-level debugging of embedded targets running Wind River Hypervisor, allowing developers to bring up virtual boards and guest operating systems.

With advanced multi-core diagnostics, you can inject errors, such as taking down a specific core, and analyze the impacts on other cores in the system to quickly spot software and hardware dependencies across multiple cores.

Cache memory management tracks cache coherency issues with main memory across multiple cores, including the ability to monitor instruction execution and identify the difference between data stored in memory versus cache. By tying the Cache view back to the editor, you can identify the source code and function for each instruction in the Cache view, improving the trouble-shooting of cache coherency problems across multiple cores.

Target Operating System Support

Workbench On-Chip Debugging supports a wide set of operating systems, including Wind River's VxWorks 5.5, VxWorks 6.3 and higher, Wind River Linux, and Linux kernels based on open source version 2.6 (for supported processors).

Workbench On-Chip Debugging provides awareness of OS objects as well as applications executing on the target, allowing debugging of complex application-kernel interactions.

Wind River Professional Services can be utilized to add support for an in-house proprietary operating system or a third-party commercial operating system. Kernel object data structure is captured using XML, enabling debug support for all defined kernel objects and their task context associated views.

VxWorks 6.x Operating System Awareness

- List of kernel tasks and object/stack summaries
- List of RTPs and object summaries
- List of RTP tasks and object/stack summaries
- List of semaphores and object summaries
- List of message queues and object summaries
- List of I/O devices and object summaries
- List of I/O drivers and object summaries
- List of I/O file descriptors and object summaries
- List of ISRs and object summaries
- List of memory partitions and object summaries
- List of triggers and object summaries
- List of watchdogs and object summaries
- List of sockets and object summaries
- List of shared libraries and object summaries

VxWorks 5.5 Operating System Awareness

- Task/process

Supported Linux Kernel Operating System Awareness

- List of processes
- List of threads
- User-mode application contexts showing application variables, stack trace, and labels

Linux user-mode debugging support is provided for debugging application code on Linux devices. Workbench On-Chip Debugging gives developers the ability to debug the Linux kernel, user applications, and shared libraries without requiring kernel instrumentation. The software also enables developers to see both system and user application contexts when debugging in Linux and eases debugging of system-application interaction issues.

For Intel targets, support is provided for debugging UEFI-compliant BIOS. This allows developers to customize, optimize, and debug the boot sequence through visibility into PEI, DXE, and HOB lists and allows support for dynamically loaded EFI modules.

Wind River JTAG Debug Units

Wind River Probe

Wind River Probe is Wind River's entry-level portable JTAG probe that supports a wide range of single-core processors. USB connectivity enables easy plug-and-play connectivity and reduces the need for an external power supply. Wind River Probe is designed to work with Wind River Workbench On-Chip Debugging and Wind River On-Chip Debugging API.

Figure 9: Wind River Probe

Key features include the following:

- USB 2.0–compliant for plug-and-play host connection
- No external power supply needed
- Adaptable to custom hardware
- User-selectable signal drive strength
- Extensible with support for ARM, ColdFire, Intel, MIPS, and Power architectures

Wind River ICE 2

Wind River ICE 2 is Wind River's high-performance, network-attached JTAG debug unit that supports a wide range of 32-bit and 64-bit single-core and

Figure 10: Wind River ICE 2

multi-core processors. Wind River ICE 2 excels in complex system debugging and multisite development. Wind River ICE 2 is designed to work with Wind River Workbench On-Chip Debugging and Wind River On-Chip Debugging API.

Key features include the following:

- Simultaneous debug of up to 16 processor cores and 32 threads
- Gigabit Ethernet interface for remote access
- Front LCD panel, easing configuration and monitoring
- Adaptability to custom hardware
- User-selectable signal drive strength
- Extensible, with support for ARM, ColdFire, Intel, MIPS, and Power architectures.
- Expandable via the optional Wind River Trace 2 external trace buffer module and Wind River Connect 2 (for supported processors)

Wind River Trace 2

The Wind River Trace 2 external trace module extends the capability of Wind River ICE 2 to include real-time trace capability for supported processors and provide better visibility into hardware/ software interaction on the target platform. This enables developers to identify and resolve the most difficult program-flow problems such as when software is randomly crashing the target or when the root cause is not easily found using standard system-level debugging

methods via register and memory access. Benefits include the following:

- 1GB trace buffer for storage of instructions and timestamp information
- Ability to capture real-time trace at clock speeds up to 200MHz
- Fast hardware-based post-processing for efficient navigation of trace buffer
- Integration with Workbench On-Chip Debugging for program-flow monitoring and user-specified trace configuration and event filtering

Technical Specifications

Host OS Support

- Fedora Core 13, 32-bit x86 and 64-bit x86-64
- Red Hat Enterprise Linux Workstation 6 (Update 1), 32-bit x86 and 64-bit x86-64
- Red Hat Enterprise Linux Workstation 5.0–5.7, 32-bit x86 and 64-bit x86-64
- Red Hat Enterprise Linux Workstation 4 (Update 9), 32-bit x86
- Ubuntu Desktop 10.04, 32-bit x86-32 and 64-bit x86-64
- SUSE Linux Enterprise Desktop 11.0, 32-bit x86-32 and 64-bit, x86-64
- OpenSUSE 11.2, 32-bit x86-32 and 64-bit x86-64
- Solaris 10, 32-bit SPARC /GTK*
- Windows XP Professional with Service Pack 3, 32-bit x86
- Windows 7 with Service Pack 1, 32-bit x86 and 64-bit x86

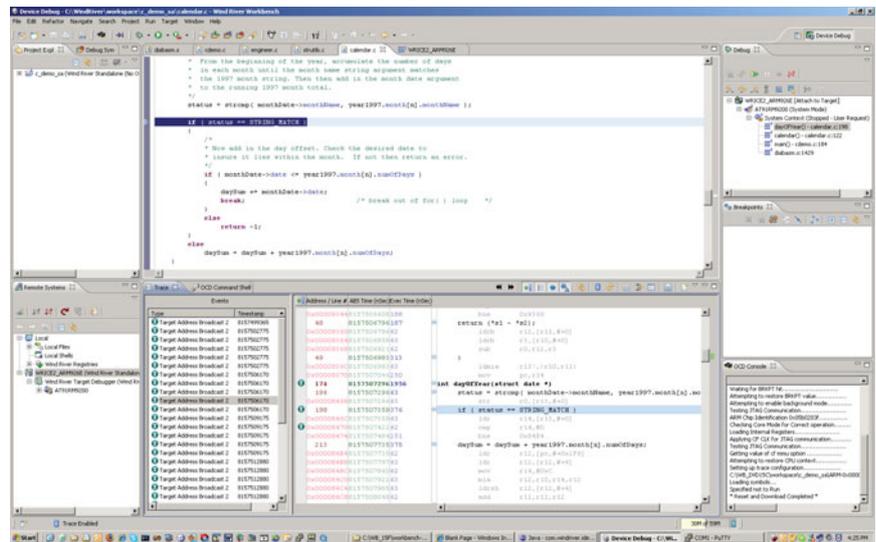**Wind River Probe is not supported on Solaris hosts.*

Figure 11: Wind River Trace 2 capture window

Target OS Support (for Supported Processors)

- VxWorks 5.5
- VxWorks 6.3 and higher
- VxWorks 653 2.3 (supported with Workbench On-Chip Debugging 3.2)
- VxWorks MILS 2.1 (supported with Workbench On-Chip Debugging 3.2)
- VxWorks Cert 6.6.4
- Wind River Linux 3.0 and higher
- Wind River Linux 4 and higher
- Linux kernels based on open source version 2.6
- UEFI-compliant BIOS and boot loaders (Intel)

A customizable target OS awareness capability for Workbench On-Chip Debugging enables support for other target operating systems to be added.

Embedded Virtualization (for Supported Processors)

- Wind River Hypervisor 1.3.1

Processor and Architecture Support

Wind River On-Chip Debugging supports a wide range of processor architectures including ARM, ColdFire, Intel, MIPS, and PowerPC. Support varies by Workbench On-Chip Debugging version and specific JTAG debug unit. For details on currently supported processors, refer to http://www.windriver.com/products/OCD/. Wind River is continually adding new processor support capabilities. If you do not see your processor listed, contact your Wind River sales representative.

Wind River Workbench On-Chip Debugging Processor Family Support

ARM	MIPS (continued)	Power Architecture (PowerPC)
ARM9	Altera MP32*	AMCC PPC403*
ARM11	Broadcom BCM11xx*, BCM12xx*,	AMCC PPC405*
ARM Cortex-A8	BCM14xx*	AMCC PPC440*
ARM Cortex-A9*	Broadcom BCM33xx*, BCM35xx*	AMCC PPC460*
ARM CoreTile Express A9x4	Broadcom BCM47xx*	CPU Tech Acalis CPU872
ARM Cortex-M3	Broadcom BCM5300x	LSI Axxia ACP3442
ST Micro SPEAr1310	Broadcom BCM53xx*, BCM5621x*,	LSI Axxia ACP3448
	BCM58xx*	Freescale PPC5xx*
ATMEL AT9x*	Broadcom BCM63xx*, BCM65xx*	Freescale MPC512x*
Cavium Econa*	Broadcom BCM70xx*, BCM71xx*	Freescale MPC52xx*
Freescale i.MX*	Broadcom BCM73xx*, BCM74xx*	Freescale MPC55xx*, MPC56xx*
Marvell*	Cavium OCTEON CN3xxx*	Freescale/IBM PPC6xx*
TI OMAP*	Cavium OCTEON Plus CN5xxx*	Freescale/IBM PPC7xx*
ColdFire	Cavium OCTEON 2 CN6xxx*	Freescale MPC74xx*
MCF52xx	NEC VR41xx*, VR54xx*, VR55xx*,	Freescale MPC8xx*
MCF53xx	VR77xx*	Freescale MPC82xx*
MCF54xx	NetLogic (RMI) AU1x00* (formerly AMD	Freescale MPC83xx*
MCF544xx	Alchemy)	Freescale MPC85xx*
	NetLogic (RMI) XLR*, XLS*	Freescale MPC86xx*
Intel Architecture	Philips PR19xx*, PR39xx*, PR44xx*	Freescale QorIQ P1xxx*
Intel Atom*	Philips PNX30xx*, PNX73xx*,	Freescale QorIQ P2xxx*
Intel Core 2* Duo	Philips PNX83xx*, PNX85xx*	Freescale QorIQ P3xxx*
Intel Core i3*	PMC-Sierra RM79xx*, RM9xxx*	Freescale QorIQ P4080*
Intel Core i5*		Freescale QorIQ P5xxx*
Intel Core i7*	Toshiba TX49xx*	P.A. Semi PA6T-1682M
Intel Xeon*	Wintegra Winpath*	ST Microelectronics SPC560xxx*
		Xilinx Virtex-II Pro X2VP*
XScale		Xilinx Virtex-4 XC4V*
Intel IXP4xx*		
Intel IXP2xxx*		
Intel IOP3xx*		
Marvell PXA*		
MIPS		
MIPS 4Kc, 4Km, 4Kp, 4KEc		
MIPS 5Kc, 5Kf		
MIPS 20Kc		
MIPS 24kc, 24kf		
MIPS 25Kf		
MIPS 74k*		
Broadlight BL23570R		

Specific processors only; for details on currently supported processors, refer to the processor support matrix at www.windriver.com/products/OCD/. If you do not see your processor listed, contact your Wind River sales representative.

Professional Services

Wind River Professional Services helps companies reduce risk and improve competitiveness. Our team delivers device software expertise within structured engagements that directly address key development challenges and contribute to the success of our clients. Our track record of timely delivery and in-depth understanding of market and technology dynamics makes Wind River a valuable implementation partner for clients worldwide. Based on our commercial-grade project methodology, service offerings include device design, BSP and driver optimization, software system and middleware integration, and legacy application and infrastructure migration.

Workbench Services

Wind River Professional Services knows how to jump-start your development efforts. Even if you opt for a non–Wind River platform, Linux distribution, host operating system, or target architecture, we can help. No matter which development environment you use, Wind River Professional Services can extend Workbench to adapt to your needs with the following offerings:

- Extend Workbench processor support
- Extend Workbench target OS support
- Validate Workbench on Linux host environment
- Validate Eclipse plug-ins
- Integrate agents

Installation and Orientation

Proper installation and orientation of Wind River Workbench On-Chip Debugging means you won't waste time solving easily avoidable problems before you can begin your next development project. Wind River offers an installation and orientation service to ensure that your project starts on time and without hassle by delivering the following:

- **Onsite installation:** Guided install on your hardware and host platform, along with a sample build process, demonstrations, and examples of customizations

- **Hands-on orientation:** Architecture, development file system, adding open source packages, porting drivers, addressing design issues
- **Advice:** Introduction to Wind River support channels and processes, additional services, project review, and consultation

The Wind River installation and orientation service will expedite your path to productivity, allow you to rest assured that we have eliminated a common source of user error, and help you realize all of the platform's potential.

Education Services

Wind River offers in-depth and comprehensive product training for Wind River Workbench On-Chip Debugging. After completing these training courses, customers can successfully design, develop, build, test, and debug device software applications in a target-host environment. Classes are offered in a variety of formats including a wide variety of regularly scheduled public courses and customer-specific onsite courses.

Wind River Personalized Learning Program is a comprehensive, systematic approach to deliver the best possible education to all developers. Unique needs of each team member are assessed, and development plans, training materials, and learning events are recommended to address work-related skill gaps. This approach results in a rapid increase in productivity.

Support Services

Wind River provides full technical support. Our global support organization is staffed with engineers who have extensive experience with Wind River products and device software development. At major support centers worldwide, our local experts can help diagnose problems, provide guidance, or answer "How do I…?" questions. Support is also available 24 hours a day at our Online Support (OLS) website (www.windriver.com/support) or by email at support@windriver.com.

Visit OLS for fast access to product manuals, downloadable software, and other problem-solving resources. Additional features, including patches and technical tips for common problems, are available for all customers upon subscription. OLS visitors can also access a community of developers to discuss their issues and experiences.

If you cannot find the information you need through OLS, contact our global support team for access to the industry's most knowledgeable and experienced support staff.

North America, South America, Asia/Pacific

support@windriver.com
Toll-free tel.: 800-872-4977 (800-USA-4WRS)
Tel.: 510-748-4100
Fax: 510-749-2164
Hours: 6:00 a.m.–5:00 p.m. (Pacific time)

Japan

support-jp@windriver.com
Tel.: +81 3 5778 6001
Fax: +81 3 5778 6003
Hours: 9:00 a.m.–5:30 p.m. (local time)

Europe, Middle East, Africa

support-ec@windriver.com
Toll-free tel.: +800 4977 4977
France tel.: +33 1 64 86 66 66
France fax: +33 1 64 86 66 10
Germany tel.: +49 899 624 45 444
Germany fax: +49 899 624 45 999
Italy tel.: +39 011 2448 411
Italy fax: +39 011 2448 499
Middle East region tel.: +972 9741 9561
Middle East region fax: +972 9746 0867
Nordic tel.: +46 9 594 611 20
Nordic fax: +46 8 594 611 49
UK tel.: +44 1793 831 393
UK fax: +44 1793 831 808
Hours: 9:00 a.m.–6:00 p.m. (local time)

How to Purchase Wind River Solutions

Visit www.windriver.com/company/contact-us/index.html to find your local Wind River sales contact. To have a sales representative contact you, call 800-545-9463 or write to inquiries@windriver.com.

Embedded Tools, Ihr langjähriger Partner für alle Wind River Entwicklungswerkzeuge
Wind River-Compiler, -Workbench, -Probe, -ICE, -ICE2