Intel® System Debugger

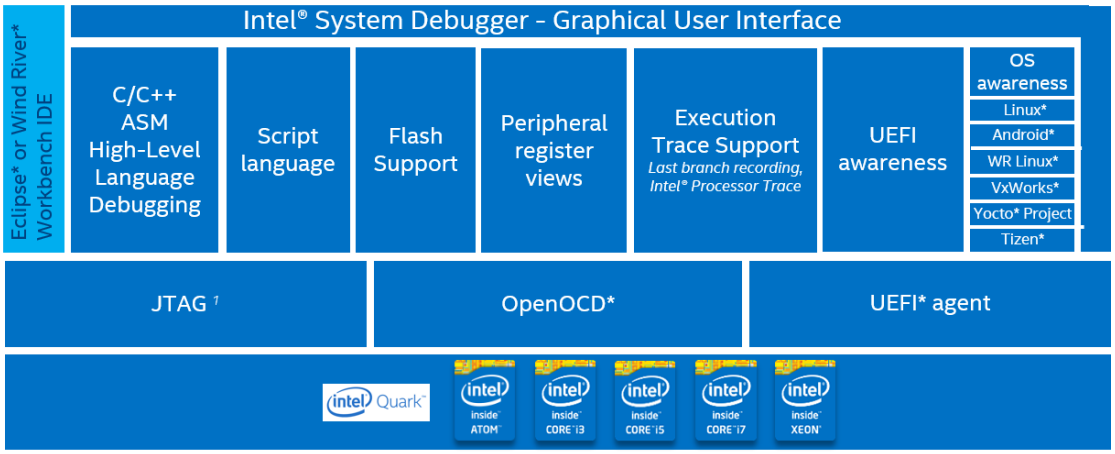
Product Brief

Deep System-wide Insight for Embedded, Mobile and IoT Systems

The explosion of embedded devices is driving an unprecedented need for efficient tools to meet shorter development cycles. The Intel® System Debugger is a sophisticated JTAG-assisted high-level-language debugger that provides deep system-wide insight into Intel® Architecture based platforms for more robustness and reliable systems. Specific OS-awareness modules for Embedded and Real-time Operating Systems make Intel® System Debugger the right solution to solve developers' complex debug challenges and help accelerate time to market of Intel Architecture-based embedded devices.

Intel® System Debugger

The Intel® System Debugger is a JTAG-based debug solution supporting in-depth debugging and tracing of Intel® Architecture-based System Software and Embedded Applications. It enables developers to debug and trace Intel® Architecture based platforms system-wide, e.g. UEFI / firmware, System-on-Chip peripheral registers, OS kernel and drivers with full OS awareness.



[1] Via Intel® ITP-XDP3 probe, OpenOCD* based devices, Macraigor* usb2demon* and EDKII* for UEFI*

Benefits:

Accelerate Time-to-Market

Speed-up debugging and testing with deep hardware and software insight

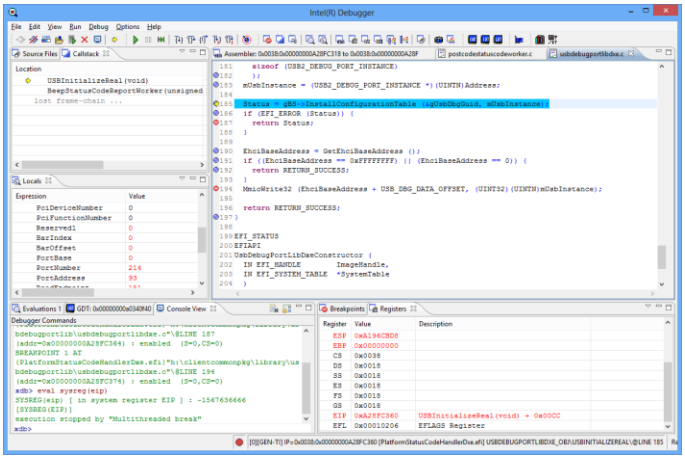
Strengthen System Reliability

Enhance system stability using in-depth system-wide debug and trace capabilities

Intel® System Debugger details

Modern User Interface

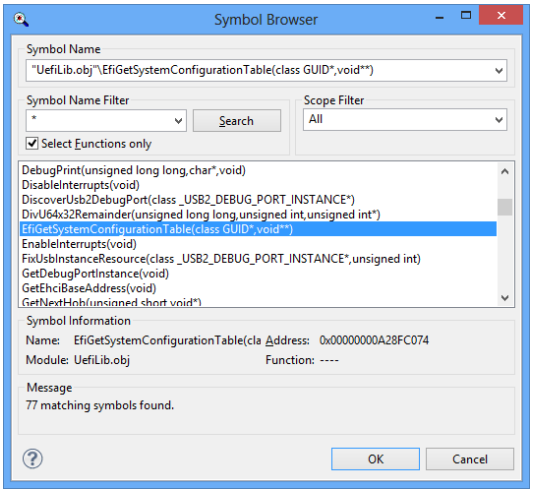
A flexible and comprehensive user interface makes accessing system status information and analyzing execution flow easy. The Intel® System Debugger provides a standalone GUI and integrates into Eclipse* and Wind River* Workbench*.



Full symbolic debug environment

More than just source code and variable names, symbols are fully integrated into the debugger. All named registers are directly usable as a symbol name. Relative math is possible, through a powerful scripting language.

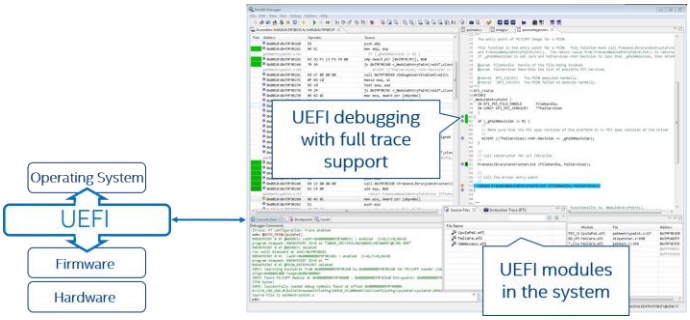
Breakpoints, evaluations, addresses all take symbolics as well as addresses.



Advanced UEFI BIOS support

Source-level debug in any phase of EFI, from reset to OS boot. Symbols can be loaded for all or selected modules.

Two modes for more productive BIOS debugging: Passive mode to inspect target memory to locate modules, load symbols. Active mode to receive notifications from an UEFI debug agent as modules are loaded or unloaded.

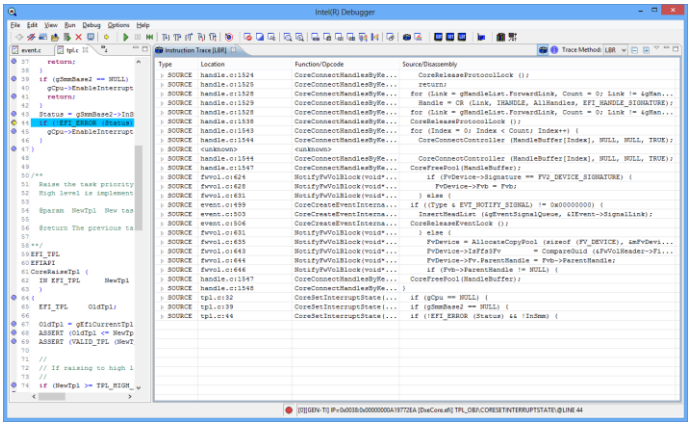


Execution Trace

Intel® System Debugger supports execution trace via:

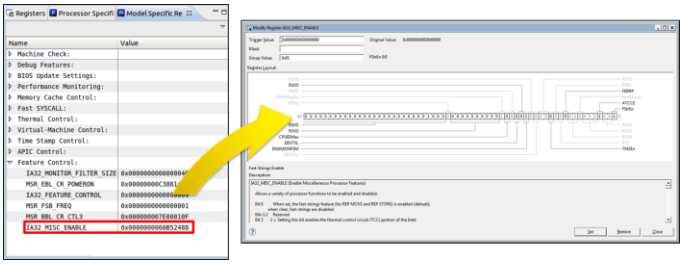
- Intel® Processor Trace (Intel® PT)
- Last-Branch Record (LBR)

Trace data is presented as C source code and the view is integrated with other source debug features. On platforms supporting Intel® Processor Trace, it offers time stamp accurate full instruction trace with configurable trace buffer depth mapped to a memory location of choice.



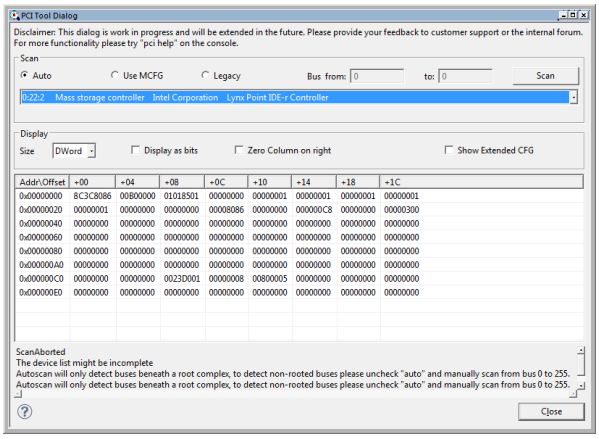
CPU State and Peripheral Registers View

GUI support for inspecting CPU state, including: Model-specific registers, architectural and processor-specific registers, system registers (PCI devices) on a case-by-case basis, and System-on-Chip peripheral registers. All registers are fully documented in the “Bitfield Editor” (on certain platforms) which make processor manuals obsolete. This feature helps to accelerate low-level driver development and validation.



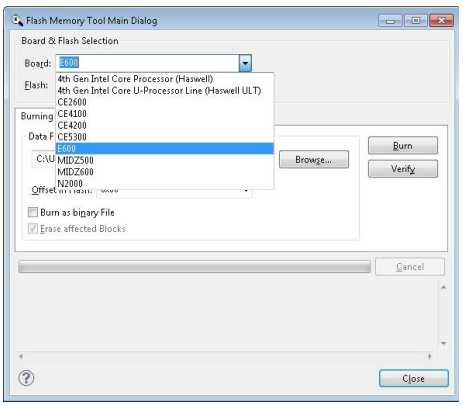
PCI Utility

Scan for devices, display device-specific registers. It provides insight into the data exchanged between the chipset and peripheral devices on the PCI bus as a given point in time, providing valuable insight for device driver developers.



Flash Programming Utility

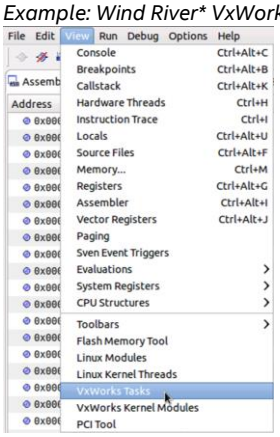
A fully integrated Flash programming capability supporting a wide range of Intel Development Platforms. Either GUI based flashing, or alternatively debugger script language driven programming enables developers to incorporate re-programming into the debug process

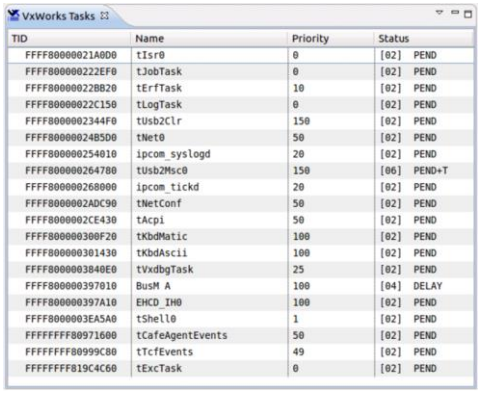


OS awareness

Incorporating kernel activities, such as kernel task lists, and loaded kernel modules into the active debug process helps to understand the system and accelerates bug fixing cycles.

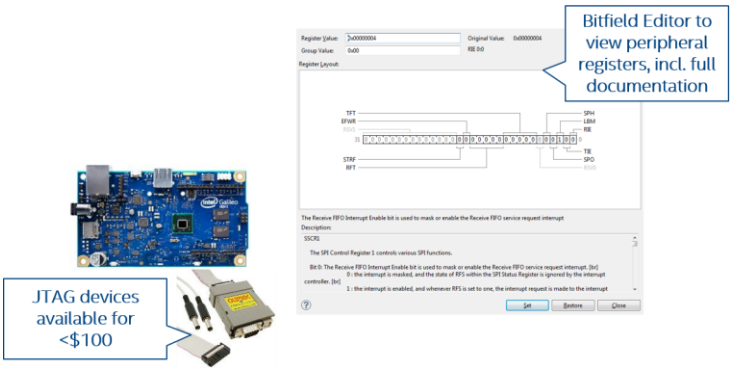
Example: Wind River VxWorks OS awareness*





System Debugging for Intel® Quark™ Platforms

Supports connection via low-cost OpenOCD*-based JTAG devices. Provides deep-insight to Intel® Quark™ SoCs



Compatibility

Host OS support

- Windows*, Linux*

Target OS support (OS awareness)

- Linux*, Wind River* Linux*, Yocto* Project
- Wind River* VxWorks*
- Android*
- Tizen*

IDE support

- Eclipse* IDE
- Wind River* Workbench*

To evaluate, go to

http://intel.ly/system-studio

and download Intel® System Studio 2015 Ultimate Edition

For more information regarding performance and optimization choices in Intel® software products, visit http://software.intel.com/en-us/articles/optimization-notice.

© 2014, Intel Corporation. All rights reserved. Intel, the Intel logo, Intel Inside, the Intel Inside logo and VTune are trademarks of Intel Corporation in the U.S. and/or other countries. *Other names and brands may be claimed as the property of others.

0214/KB/SPC 335558-0555US