PRODUCT PROFILE

SourcePoint™ for ARM® Architectures

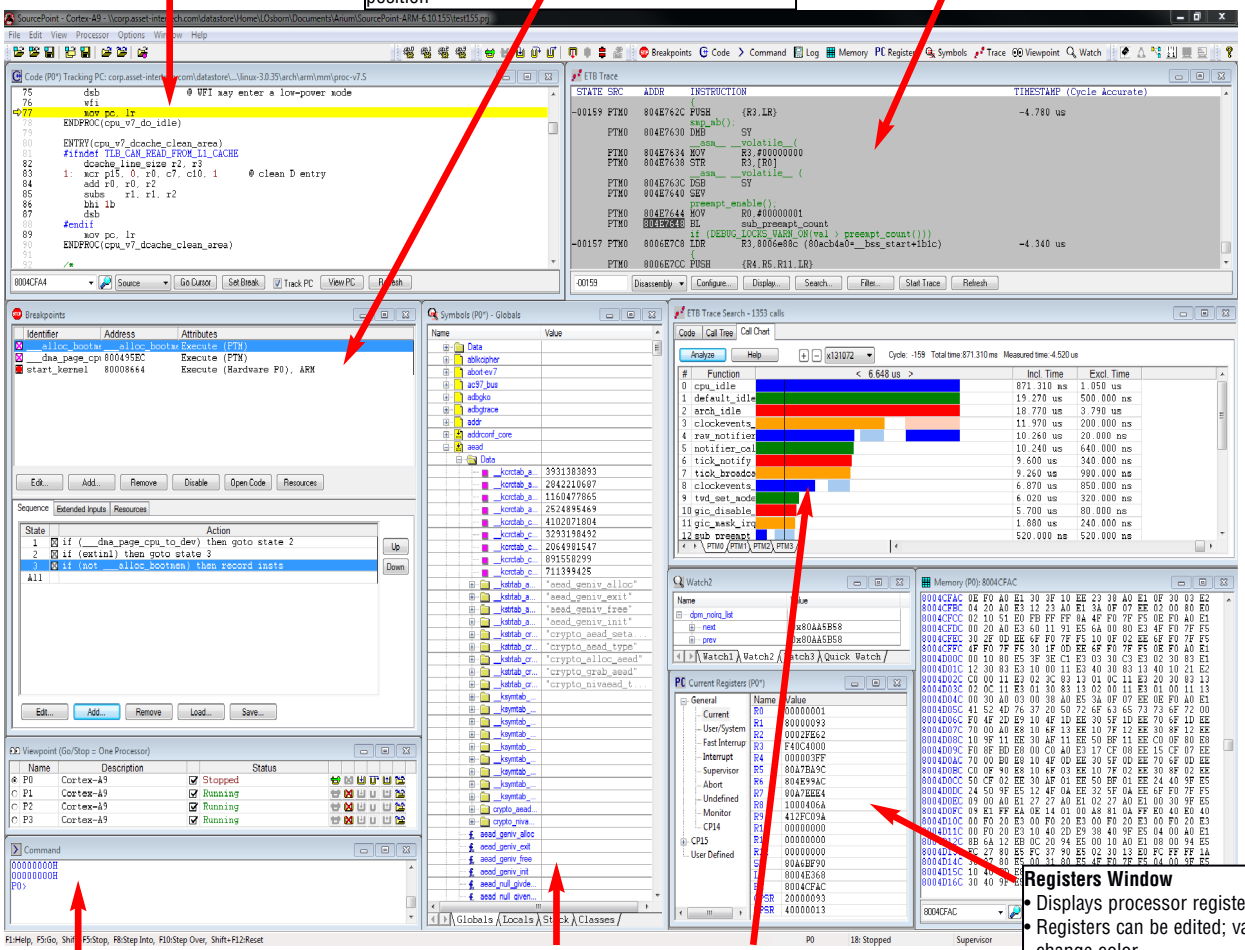
SourcePoint™ for ARM is ASSET InterTech's flagship software debugger for ARM®, Freescale i.MX™, TI OMAP™ and ARM core processors/FPGAs. The software, versatile, customizable, and reliable, offers excellent visibility to C and C++ code and its execution. It runs on Microsoft® Windows® and Linux hosts, debugging platforms with various RTOSs, including Linux, or no RTOS at all.

SourcePoint for ARM is a hardware-assisted debugger and Arium debug probes and trace port analyzer (TPA) provides the target connection. Together they deliver feature-rich, real-time debug, from the way SourcePoint views are required to the way the command language scripting works, to the speed with which tasks execute. Run-control is state of the art. Trace is integrated. Performance analysis data match the real execution time of the program being run. There's multi-core debug, too, complete with individual processor control.

Code Window
- Displays C and C++ source, assembly code, comments, symbols, and breakpoints
- Display modes easily selected
- Step in C and/or assembly code
- Set breakpoints from this window
- View data values

Breakpoints Window
- Set, edit, remove, enable, and disable breakpoints
- Set complex,multi-level sequences
- View hardware decodes and resources
- Lets the developer select pre-center - post trigger position

Trace Window
- Provides record of real-time ETM events; data can be used to determine exact path of code execution
- In multi-processor environments, disassembled code displays using different color for each processor



Command Window
- Runs robust C-like command language for run control, loop execution, data and array variable use, file I/O access, and more
- Lets the developer write sophisticated macros for testing or set up

Symbols Window
- Access to all symbols and source code
- Composite variables, including arrays, structures, and unions, expandable to show their subelements

Trace Search Call Chart
- Call Tree/Chart Displays function execution call stack
- Function execution precidence with cycle accurate time stamps

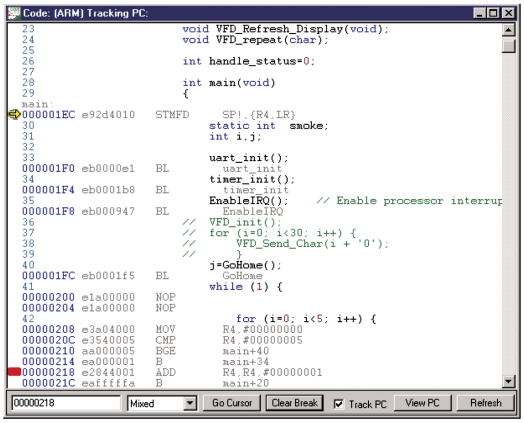
Registers Window
- Displays processor registers
- Registers can be edited; values change color
- Flyovers show labeled bit fields
- Allows user-defined register groups

ASSET

SourcePoint™ for ARM® Architectures

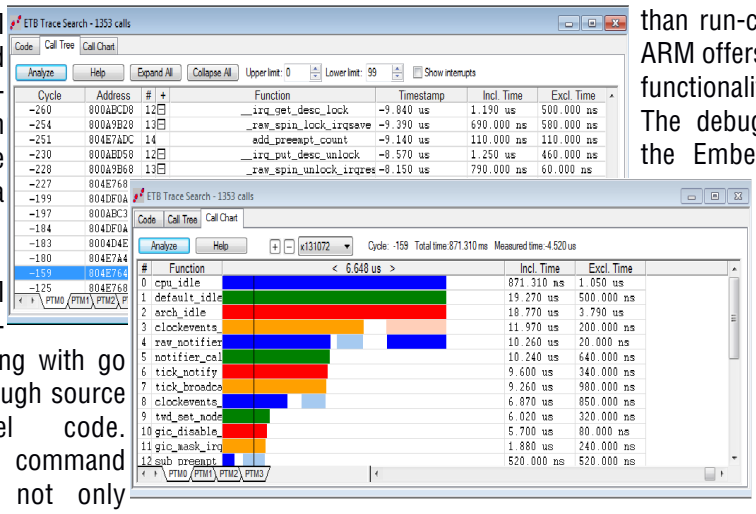
SourcePoint for ARM offers a number of unique but highly intuitive features, making it and exceptional debugger for today's ARM-architecture projects. A good debug solution offers the programmer or developer layers of debugging options - from the simple breakdown of code into ever smaller iterations to the ability to capture and analyze huge amounts of execution history in a single run. To meet developers' needs, SourcePoint for ARM debugger offer several solutions, of run-control or run-control with integrated trace.

Code Window
- Displays C or C++ source or assembly code, or allows the developer to see both; also displays comments, symbols, and breakpoints
- Allows single stepping (in C or ssembly code)
- Offers breakpoint setting from this window
- Makes register or variable values visible via flyover help
- Works with multi-processor systems via multiple Code windows



Managing Run-Control

The key to a successful run-control debug strategy lies in the ability to set accurate breakpoints and step through code. SourcePoint for ARM offers processor and soft breaks via simple GUIs. Breaks can also be set from the Code window or a command line.



SourcePoint for ARM uses the usual stepping commands along with go and halt to step through source or assembly-level code. SourcePoint's C-like command language includes not only industry-standard run-control commands, but lets the developer execute loops, use data and array variables, access file I/O, and more. Unlike some command languages, SourcePoint is intuitive; developers do not need to know a two letter code for each command.

Intuitive windows can be opened to view the state of the processor and make modifications to values, including Symbols windows, Registers windows, Memory windows, and user-defined Watch windows. In multi-core environments, developers can mask processors together or individually and start, step, and stop those not masked.

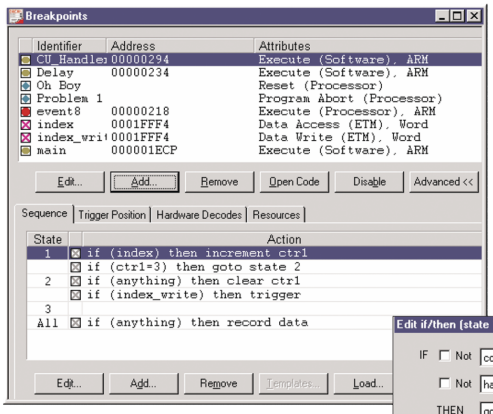
ASSET's debug solutions are designed with time in mind. Whether downloading files or images, stepping through code, or coming back after hitting stop, the event executes with incredible speed.

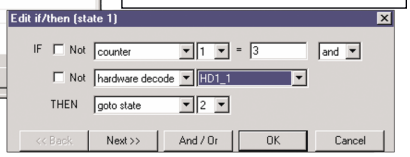
Capturing, Filtering Analyzing Execution History

For developers who want more than run-control, SourcePoint for ARM offers some of the best trace functionality on the market today. The debugger handles trace via the Embedded Trace Macrocell (ETM) in ARM processors. Several features make SourcePoint for ARM a true "solutions" debugger, including integrated trace, trace buffer depth, easy multi-level triggering, and performance analysis.

Integration. SourcePoint for ARM trace is integrated into the software package; it is not an "add on." Advantages? There's the obvious - developers do not have to deal with (or pay for) a separate trace port analyzer. There is no new software to load. Users do not have the integration problems that plague add-on solutions. Apart from the obvious, the greatest advantage is being able to open a Code window and scroll through the collected trace, examining the actual code and how the compiler compiled it, with correlated code and trace.

SourcePoint™ for ARM® Architectures





Complex Triggering

- User-friendly dialog boxes let the user set/edit multilevel if/then clauses to create complex triggering sequences
- Unlike many other applications, the user does not need to know the details of the ETM registers when programming sequences

Deep/fast trace. With the Arium LX-1000 for ARM trace port analyzer, SourcePoint for ARM delivers trace depth up to 8 GBytes with a capture rate up to 680 MHz.

Multi-level triggering. When used with the Arium LX-1000 for ARM and a target with ETMv1 or ETMv3, SourcePoint for ARM offers superb complex sequencing via a series of user-friendly GUIs. Based on breakpoints and other user-defined events, trigger parameters can be qualified or refined. For example, an event may occur thousands of times in a program. A developer can set up a breakpoint to trigger only after the first thousand times it occurs and only if it occurs after a particular address. Unlike other debuggers, SourcePoint for ARM does not require "rocket science" to set up the triggering sequence.

Performance analysis. SourcePoint for ARM includes performance analysis for use with ARM cores with ETM. While most development tools rely on a compiler to handle performance analysis, SourcePoint uses the ARM ETM to mine data from code. The ETM has up to 16 address comparators that can be programmed with function entry and exit points.

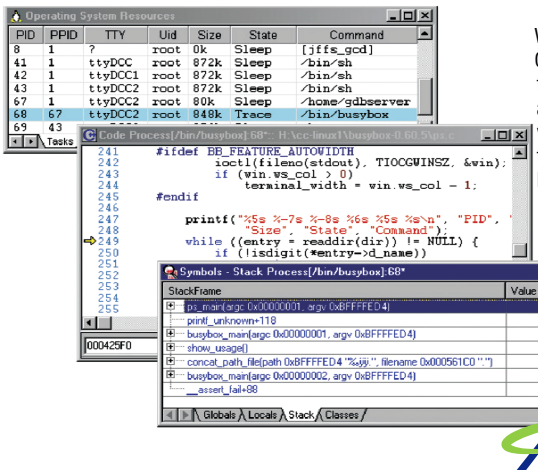
Trace qualification is used to record only those addresses in the trace buffer. This gives the developer approximately 140,000 real-time entry and exit points in a 1 MB trace buffer (each address generating a 5-byte broadcast address consuming 7 cycles), resulting in a single trace capture that can contain thousands of instances of each address range. This solution requires no changes to the user's program, and measured performance data match the real execution time of the program being run.

SourcePoint for ARM also offers code profiling for targets with processors without ETM.

SourcePoint on a Linux Platform

SourcePoint delivers industry-leading functionality for developers working on Linux-based embedded systems, including:

- Full symbolic, source-level debugging of Linux kernel code
- Source-level debugging of Linux embedded applications
- Launch of or attachment to processes with seamless transitions to and from the kernel and each process
- Code debug on initial target bringup immediately from board reset
- Dynamically loaded module debug (insmod)
- Threaded application debug support
- Linux shared libraries debug
- Specialized breakpoints to stop the execution of a process without stopping the processor or causing it to enter debug mode
- Flash programming for kernel and file-system download
- Linux console hosting devices from within SourcePoint, eliminating the need for a serial port or video device on the target and simplifying the debugging of "headless systems"



When debugging a Linux OS-aware target, clicking on the Task tab brings up the attendant code in the Code window and makes available for display associated symbols in the Symbols window.

SourcePoint™ for ARM® Architectures

• Intuitive, Windows-like debug environment

SourcePoint for ARM allows concurrent debugging of Linux kernel code and Linux application processes. Within SourcePoint, two views provide the interfaces to Linux-aware debugging features. The Operating System window lists Linux processes and serves as the primary interface for task debugging. The Target Console window emulates multiple terminals that serve as the Linux system console and as the standard input and output device for processes launched for debugging.

Shortcuts within shortcuts

SourcePoint for ARM incorporates hundreds of options, commands, and functionalities designed to spur the debug process forward. Windows are designed to be intuitive. They can be docked, floated, or minimized. Commands are available from multiple locations - menu bars, icon bars, context menus, a command line. Symbols and their values are easy to find and change.

Items are grouped logically in intuitive windows and dialog boxes. For example, target configuration options exist under a single view. From the dialog, the memory map of the target can be defined, the type and address range of flash memory devices declared, and target flash operations performed. Target configurations can be loaded from a developer's target database file and saved to SourcePoint for ARM and/or the target database file.

SourcePoint for ARM offers a number of user-defined options. These include windows that allow definition of memory mapped I/O devices and related registers and areas of memory in one view. Developers can keep track of multiple devices without having to keep multiple views on their screen.

Outstanding Support

ASSET InterTech offers exceptional service and support for all of its debug solutions. Highly qualified technical support staff are available during regular working hours and delays getting to them are minimal. Often they can pinpoint a problem immediately or on review of a dump of the log and project files. Support staff can also troubleshoot particularly difficult problems via WebEx™, an online interactive solution that allows them to see a developer's code but lets the developer control the session. Additionally, there are downloads and technical documentation available on the ASSET InterTech web site.

For more information, contact your sales representative or ASSET InterTech tools distributor or visit www.asset-intertech.com or go directly to the SourcePoint for ARM web page.

Hardware

Arium LX-1000e for ARM Trace Port Analyzer with Serial and Parallel Trace

The Arium LX-1000e for ARM delivers a trace depth of 2 GByte upgradeable to 8 GBytes. The trace capture engine traces instructions/data at rates up to 680 MHz and is wide enough to support up to 6 lanes of serial trace packets up to 6.25 GHz. The Arium LX-1000e for ARM represents the fastest trace port analyzer on the market, giving ASSET customers a valuable investment to support future devices.

Arium LC-500Se Run-Control Probe

The Arium LC-500Se offers reliable run control and a number of unique features, including a single window for viewing memory-mapped I/O devices and a JTAG rate of up to 20 MHz. Packaged with ASSET's flagship SourcePoint for ARM software interface, the run-control probe supports full ARM and Thumb™ instruction sets and is designed for use with today's most popular compilers.

17791 Fitch Avenue, Irvine, CA 92614 Voice: 949-517-4150 outside the US Fax: 714-731-6344 E-mail: ai-info@asset-intertech.com Web: www.asset-intertech.com

D0130L-2014