

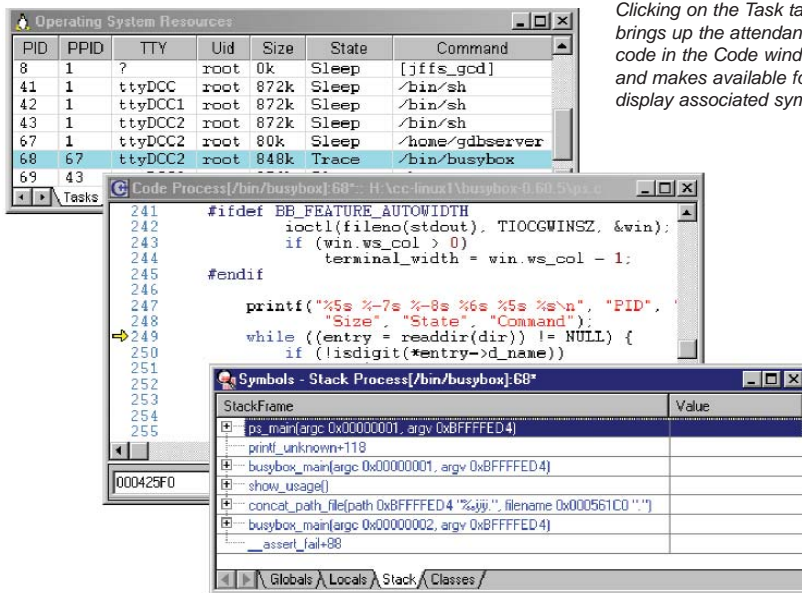
LINUX DEBUG USING SOURCEPOINT™

AMERICAN ARIUM INTRODUCES THE FIRST DEBUGGER FOR LINUX THAT SEAMLESSLY DEBUGS BOTH THE KERNEL AND PROCESSES AND THEIR INTERACTIONS WITH ONE SET OF TOOLS. USERS CAN DIRECTLY LAUNCH FROM OR ATTACH TO ANY TASK FROM THE DEBUGGER INTERFACE. PROCESSES AND THEIR STATES ARE VISIBLE IN A SINGLE TABLE; USERS CAN ACCESS ASSOCIATED

SOURCE CODE SIMPLY BY CLICKING ON A LISTED PROCESS.

VISIBILITY IS IMMEDIATE FROM BRING-UP OR RESET! A CONSOLE WINDOW IS AVAILABLE THROUGH THE JTAG. NO MORE WAITING UNTIL A SERIAL OR ETHERNET PORT IS "LIVE." NO MORE WAITING FOR THE KERNEL AND CONSOLE DRIVERS TO COME UP. NO MORE WAITING, PERIOD.

ARIUM'S HARDWARE-ASSISTED SOURCE POINT™ DEBUGGER IS AMERICAN ARIUM'S FLAGSHIP SOFTWARE DEBUGGER FOR ARM® AND INTEL XSCALE® PROCESSORS. THE TOOL, VERSATILE, CUSTOMIZABLE, AND RELIABLE, OFFERS EXCELLENT VISIBILITY TO C AND C++ CODE AND ITS EXECUTION.



Clicking on the Task tab brings up the attendant code in the Code window and makes available for display associated symbols.

Features and Benefits

- **Kernel debug straight out of reset - an industry first!**
- **Dynamic debug of loaded Linux kernel modules - another first!**
- **Application and process debugging with Linux OS awareness for a complete debug solution in one package.**
- **Linux console output through JTAG port eliminating serial or ethernet port requirement**
- **Stack trace viewing for quick context information.**
- **User friendly ARM ETM trace configuration and multi-level triggering**
- **Comprehensive, flexible code editing with source navigation/browsing; code can be debugged "on the fly" with SourcePoint IDE™.**

The Linux-aware features of SourcePoint provide a number of important capabilities for users who are working on Linux-based embedded systems. Features include:

- Full symbolic, source-level debugging of Linux kernel code.
- Source-level debugging of Linux embedded applications, including the ability to start or stop a Linux process, attach to a process, view source and symbols for a process, and set breakpoints within a process.
- Launch of or attachment to processes with seamless transitions to and from kernel and each process.
- 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.
- Debug of dynamically loaded kernel modules with full source available at init ("busybox" version of "insmod" command).

LINUX DEBUG USING SOURCEPOINT

- 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.

SourcePoint allows concurrent debugging of Linux kernel code and Linux application processes via two new GUIs. 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.

SourcePoint utilizes the ARM Debug Communication Channel (DCC) hardware to communicate over the JTAG port with code on the target, eliminating the need for any serial or network hardware as a prerequisite for debugging. As a result, the console is available from power-up without any other hardware dependencies except the processor core and the JTAG debug port.

MANAGING RUN CONTROL

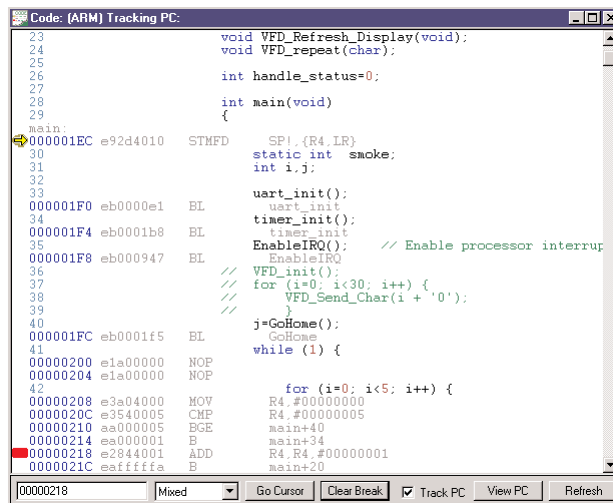
The key to a successful run control debug strategy lies in the ability to set accurate breakpoints and step through code. SourcePoint offers processor breaks and unlimited soft breaks via simple GUIs. Breaks can also be set from the Code window or a command line.

SourcePoint 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 excellent 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.

One of the newer features of the software is a handy symbols finder that displays any program symbol and its memory address. A dialog can be sum-

Code Window

- Displays C or C++ source or assembly code, or allows the user to see both; also displays comments, symbols, and breakpoints
- Allows single stepping (in C or assembly 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



```
Code: (ARM) Tracking PC:
23 void VFD_Refresh_Display(void);
24 void VFD_repeat(char);
25
26 int handle_status=0;
27
28 int main(void)
29 {
30     main:
31     000001EC e92d4010 STMFD   SP!, {R4, LR}
32     static int smoke;
33     int i,j;
34     uart_init();
35     timer_init();
36     EnableIRQ(); // Enable processor interrupt
37     VFD_init();
38     for (i=0; i<30; i++) {
39         VFD_Send_Char(i + '0');
40     }
41     j=GoHome();
42     while (1) {
43         for (i=0; i<5; i++) {
44             00000200 e1a00000 NOP
45             00000204 e1a00000 NOP
46             00000208 e3a04000 MOV     R4,#00000000
47             0000020C e3540005 CMP     R4,#00000005
48             00000210 aa000005 BGE     main+40
49             00000214 ea000001 B       main+34
50             00000218 e2844001 ADD     R4,R4,#00000001
51             0000021C ea000001 B       main+20
52     }
53 }
```

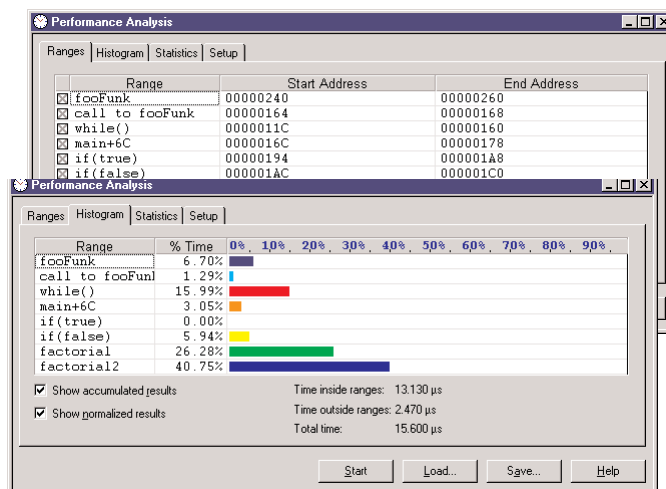
moned by selecting a program from within a Symbols window Global tab and pressing CTRL-F, or by pressing CTRL-S from anywhere within SourcePoint.

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.

Arium'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.

Performance Analysis Window

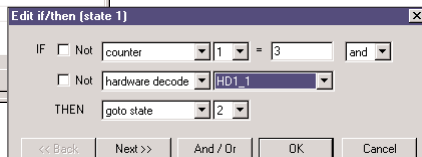
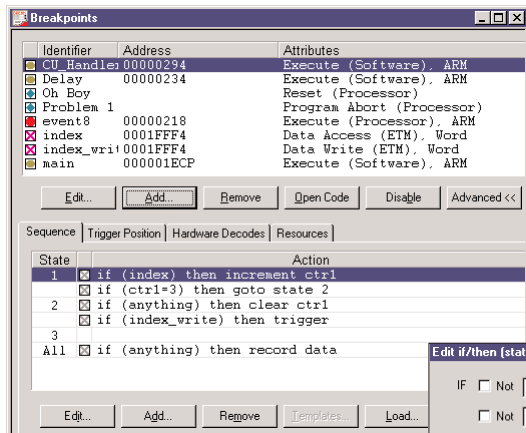
- Helps find bugs, profile functions, and gain better understanding of code coverage
- Setup is easy and intuitive
- Not limited to functions; any address range may be specified
- Not limited to specific compilers; no need to recompile code
- Does not alter the code or timing of experiment



CAPTURING, FILTERING, ANALYZING EXECUTION HISTORY

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

LINUX DEBUG USING SOURCEPOINT



Complex Triggering

- User friendly dialog boxes let the user set/edit multi-level 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

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.

Arium also offers code profiling for targets without processors with ETM. Code profiling measures the amount of time a processor spends in an address range; SourcePoint uses periodic sampling of function code to gather the data.

Integration. SourcePoint 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.

Deep trace. With Arium's SC-1000A emulator, SourcePoint offers trace depth up to 4 M samples, depending on the application. A triggering mechanism in the SourcePoint Breakpoints window allows users to determine where in the run the fill begins. In addition, an Arium proprietary trace buffer compression technology can be enabled to increase samples up to 15 times standard ETM trace depths to provide up to 60 M samples of trace.

Multi-level triggering. When used with the SC-1000A and a target with ETM, SourcePoint 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 user 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 does not require "rocket science" to set up the triggering sequence.

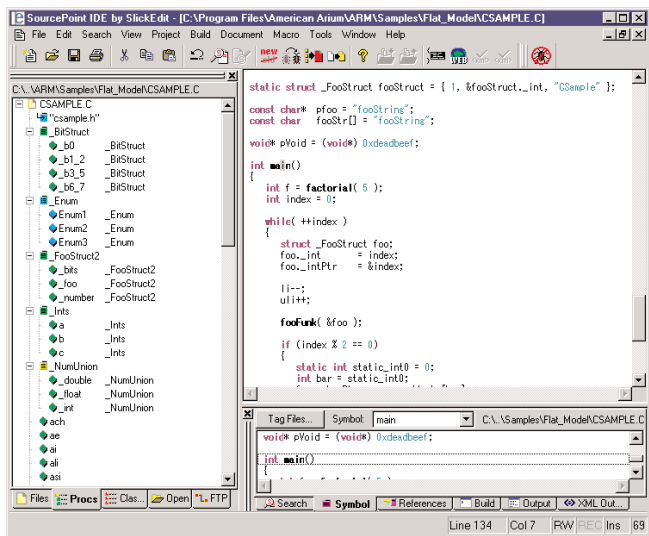
Performance analysis. SourcePoint includes performance analysis for use with ARM9™ cores. While most development tools rely on a compiler to handle performance analysis, SourcePoint uses the ARM ETM to mine data from code. The ETM has up to 16 address comparators that can be programmed with function entry and exit points.

Trace qualification is used to record only those addresses in the trace buffer. This gives the software designer approximately 140,000 real time

WORKING WITH SOURCEPOINT IDE

SourcePoint offers source code editing "on the fly" through SourcePoint IDE (Integrated Development Environment), powered by Visual SlickEdit®, that ships under a separate license on the SourcePoint CD. The tight coupling of the debugger and editor allows the developer to move back and forth between them with ease, seamlessly editing source code, recompiling it, loading and running it on the target, and debugging it again.

SourcePoint IDE offers full project management support including project and make-file generation and build, an



SourcePoint IDE Window

- Quick and easy movement back and forth between debugger and editor
- Includes full project management support
- Easy integration of third-party development tools (e.g., compiler and linker)
- Ability to view differences between source files, directories, source trees, or symbols
- Multiple file and directory search and replace functionality

LINUX DEBUG USING SOURCEPOINT

ARIUM HARDWARE

SC-1000A Emulator

The SC-1000A emulator, which supports the ARM Embedded Trace Macrocell (ETM) and Intel



XScale real time trace, features first-of-its-kind trace compression technology that increases trace depth exponentially. Coupled with superb code manipulation and real time event management, the SC-1000A offers exceptional functionality of and superior visibility to embedded software.

LC-500 Emulator

The LC-500 offers outstanding 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 Arium's flagship SourcePoint software interface, the debug solution supports full ARM and Thumb™ instruction sets and is designed for use with today's most popular compilers.

SHORTCUTS WITHIN SHORTCUTS

SourcePoint 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 user's target database file and saved to SourcePoint and/or the target database file.

SourcePoint offers a number of user-defined options. This includes a window that allows definition of memory-mapped I/O devices and related registers and areas of memory in one view. Users can keep track of multiple devices without having to keep multiple views on their screen.

OUTSTANDING SUPPORT

American Arium 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 lets them see a developer's code but lets the developer control the session. Additionally, there are downloads and technical documentation available on the Arium Web site.

SourcePoint and SourcePoint IDE ship with Arium hardware or separately as upgrades to previously purchased hardware-assisted solutions. For more information, contact your sales representative or Arium tools distributor or visit our Web site at www.arium.com.

easily configurable environment, direct invocation of SourcePoint from within SourcePoint IDE with current project files loaded and ready for debug, and easy browsing for tool error output.

Editor features include syntax expansion and indenting, aliases, path/file-name completion, symbol listing and replacement, Context Tagging™, auto list members/parameters, auto parameter information, tag preview, and source code navigation. SourcePoint IDE also supports a class browser, version control management, and workspaces and projects supporting multiple configurations.

Linux Kernel Versions Validated

2.4.18, 2.4.20, 2.4.21, 2.5.30
MontaVista 3.1, Timesys 4.1

Required Files (Arium Supplied)

gdbserver (requires gdb 6.0) - for application debug
ariumdcc - for console via JTAG
dccwrap - for console via JTAG



PO Box 7054, 5980 AB
Panningen, The Netherlands
Tel: +31 77 307 8438
info@logic.nl www.logic.nl

