ZAP Cross Debuggers for Freescale Microcontrollers

***ZAP** is a family of full-featured C and assembly language source-level debuggers designed to give Freescale embedded microcontroller developers a consistent and productive debugging environment across multiple target processors.*

Key Features:

Provides a Portable Debugging Environment,

Debugger for Each Stage of Your Project,

C and Assembler Source-level Debugging,

ANSI C Debugging,

Array and Structure Explorer,

Debug Fully Optimized Code,

Easy-to-use Graphical User Interface,

Extensive Program Control & Analysis Features,

Graphical Performance Analysis,

Code and Data Coverage,

C Level Trace,

Robust Script language,

Automated Testing,

Real-Time BDM debugging,

Hardware Breakpoint support,

FLASH and EEPROM Programming,

Debug from FLASH,

ZAP Gives You a Portable Debugging Environment Across Projects

If you now use or are looking to design-in different Freescale microcontrollers into your embedded projects, ZAP helps you standardize your debugging tools across projects by giving you a portable graphical user-interface which helps improve developer productivity, saves expensive retraining costs and reduces product time-to-market pressures. ZAP is available for most Freescale microcontroller families and across target debugging hardware, so you have the freedom to choose a development environment that meets your needs and budget.

ZAP Addresses Your Debugging Needs At Each Stage of Your Project

ZAP's multiple configurations are designed to address your debugging needs as your project moves from the design stage to final integration and test; ZAP configurations supported are: **software simulation, target monitor, MON08, background debug mode**, and **in-circuit emulator**. Each configuration gives you essentially the same user interface, thus vastly improving your productivity, but each addresses a different stage of your project.

C and Assembler Source-level Debugging

If your source code is written in C, you want to debug at the C level; if parts of your source code are written in assembly language you want to debug at the assembly language level. ZAP automatically supports both modes without any special options or settings, so you always see your original source code in the Source window. If you are debugging at the C level, you can activate a Disassembly window that shows you the corresponding assembly language code for each line of C source.

ANSI C Debugging

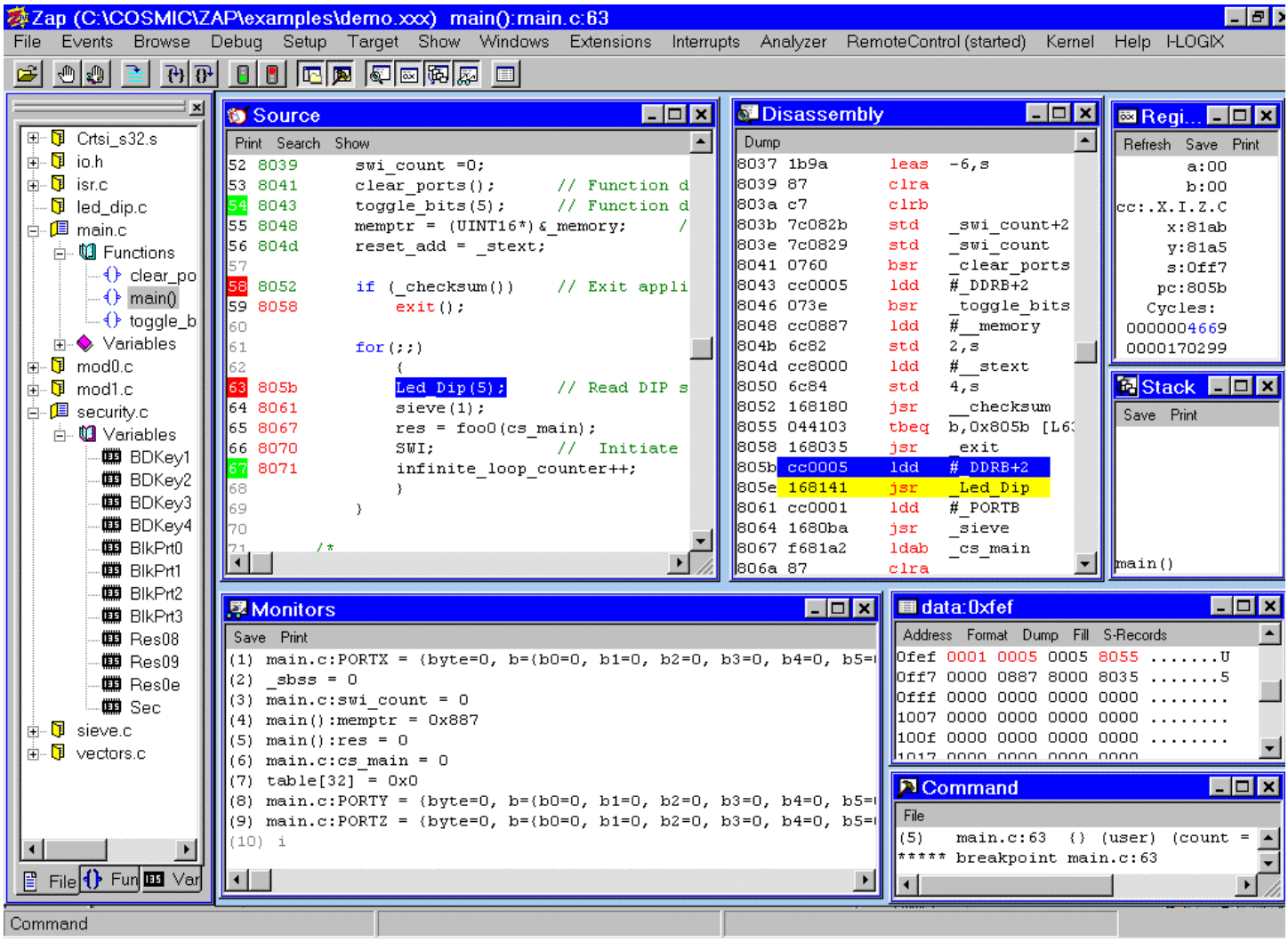
ZAP provides point and click access to any C object or construct including enums, bit fields, strings, doubles/floats, structures and stack frames. View objects and memory in various formats including character, binary, octal, hexadecimal, unsigned, ASCII and string formats.

ZAP Debugs Fully Optimized C Code

COSMIC C Compilers generate exactly the same code with and without the debug option, which means you can debug fully optimized C code and, once debugged using ZAP, your code is ready for PROM burning with no need to recompile. Several versions of ZAP even program FLASH and EEPROM directly. Debug information is never downloaded, FLASHED or PROMED into the target system.

Easy-to-use Graphical User-Interface

ZAP is a full featured C and Assembly source level debugger interface available on PC for Windows 95/98/NT/2000 and XP. ZAP is an easy to use, intuitive debugging environment with a serious debugging engine incorporating over 15 years of embedded compiler and debugger technology. Zap includes all of the bells and whistles that desktop C programmers expect along with all of the low level features assembly language programmers need. ZAP's core technology along with it's look and feel are available for many different target processors and execution environments including Simulation, Monitor, Background Debug Mode and in-circuit emulation.



ZAP Main Window

The main ZAP window acts like a desktop containing all of the display windows as well as all of the setup menus and status information. This window includes an optional button bar to control execution and displays. All of the buttons, menus and configuration are the same for all ZAPs except for the Target menu, which contains any target specific options and features such as FLASH programming, Hardware breakpoints, emulator trace etc..

Source Window

The Source window displays the source code for the loaded application and maintains the display during execution with the current Program counter. This window is color coded for C and Assembly keywords and optionally displays absolute addresses or code coverage information. You can double click on line numbers to set breakpoints, select variables for the monitor/watch window and single step the program.

Disassembly Window

The Disassembly window reads from the target's memory to provide an accurate disassembly of the actual programmed code. Colored highlights in the disassembly window are aligned with the source window to show the assembly instruction(s) corresponding to the highlighted lines in the source window. This window is useful when single stepping at the assembly level or tweaking the performance of the code. You also can double click on the addresses to set breakpoints.

Register Window

This window displays the internal CPU registers and allows you to modify the values by double clicking. The Simulator version of ZAP also includes two cycle accurate MCU counters. One displays the total number of cycles executed since the last reset and the second displays the number of cycles from the last execution command.

Command Window

The optional Command window is used to load command scripts or type any individual commands. The command window is useful for monitoring individual structure and array members as well as creating automated debugging sessions

Stack window

The Stack window displays a list of the functions in the current stack frame in the order they were called. It also displays the value of each function's arguments. This display provides a convenient way to follow execution through nested and recursive function calls.

Monitor/Watch window

The Monitor window is a scrollable window, which displays the values of monitored data objects and expressions. This window is updated each time the execution environment halts execution. You can double click on any variable to change its display format or update its value. Supported formats include: address of, character, binary, octal, decimal, unsigned, hexadecimal and string. Depending on the hardware used, ZAP also supports **Real-Time Monitors** that allow you to monitor and update variables and memory while the program is executing.

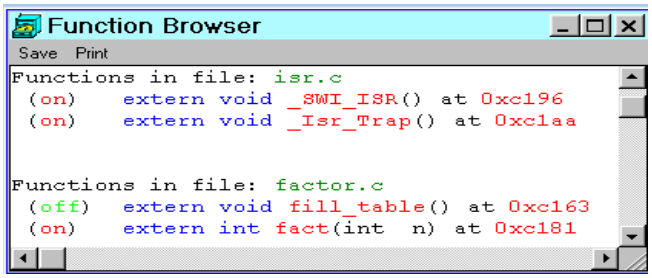
Data window

The Data window displays a block of target memory in a variety of formats including: disassembly, byte, word, long, binary, octal, decimal, hexadecimal and ASCII. You can change R/W memory simply by clicking on the value and typing the new value over the old. Changes in memory contents are highlighted to make it easy to track memory modifications as your program executes.

- **Fill Memory** – This option allows you to fill memory between addresses with a specific or random pattern. The fill pattern can be a byte, word or long.
- **Upload Data** – The data window can also be used to upload data from target memory to a text file. The data can be formatted as a data dump, disassembly or uploaded and converted to **S-Records**. This is useful for extracting an existing application from a target system.

Source Browsing

The source browser lets you search through the application's source files and double click on any function or file name to open multiple windows to let you view any source file in your application. You can also set breakpoints in any source browser window without changing the main source window.



ZAP Configurations

ZAP configurations define the code execution engines that ZAP uses to execute your application code. In general, each configuration is a separate application that provides full C and assembly language debugging. However, in some cases certain features are absent due to limitations in the target execution engine. For each target microcontroller supported by COSMIC products, there are at least two separate ZAP packages: (1) a remote, real-time debugger and (2) a hosted, non real-time simulator/debugger. All ZAP configurations include the standard windows described previously as well as the following features:

- **Simulated I/O** – The simulated I/O feature is a very powerful feature that allows your application to interact with multiple data files located on the host system. This is typically achieved via a command script using watch points at PORT addresses along with special fopen, fread and fwrite ZAP commands.
- **Automated Testing** – ZAP offers a robust high level command and scripting language which can be used to load multiple command scripts with multiple applications and perform any combination of ZAP commands without any user interaction. Compatible with most unit test programs including Vector Software's VectorCast.
- **Save Layouts and Sessions** – This feature allows you to save a debugging session and play it back later. The save layout command saves the size and location of any open ZAP display windows. The save session command saves the layout, loaded application, source path and any monitor and data windows with their contents. In addition, ZAP can optionally save the

current layout on exit as the default for the next time ZAP is opened.

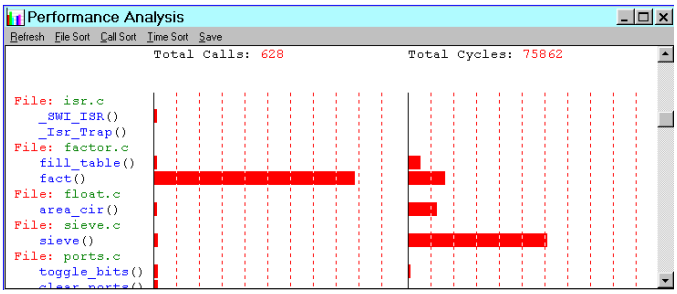
Simulator Configuration (ZAP SIM)

This configuration of ZAP provides an integrated **cycle accurate** instruction set simulator as the execution engine. Simulation is useful in the initial stages of your design, before you have access to target hardware, because it gives you a convenient way to debug algorithms and data structures before moving onto the more complicated hardware and software integration phase. You can also profile the execution of your code with the built in cycle accurate instruction timer to help you tune your application. The simulator configuration of ZAP is available for all target microcontrollers with Cosmic compiler support. ZAP SIM includes all of the standard ZAP features as well as the following additional features::

- **C and Assembly trace** (non-real time) – step backwards and forward through a recorded trace complete with recorded monitor values and register values.
- **Interrupt Simulation** - Simulate interrupt using the cycle accurate timer as a trigger mechanism or trigger on execution of an address.

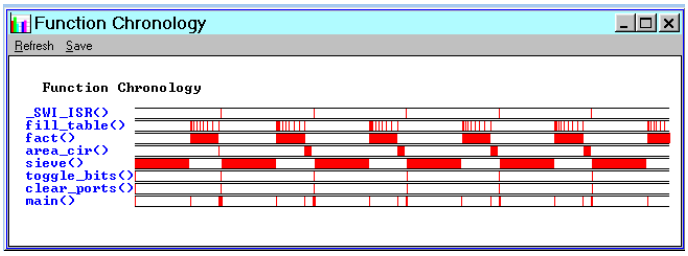
Graphical Performance Analysis

ZAP SIM includes a performance analysis feature that displays cycles of execution on a file by-file or function-by-function basis. Double click on a function name or file name to display the number of times the function was called and the total number of cycles executed for that function. Performance information may also be saved as a text report.



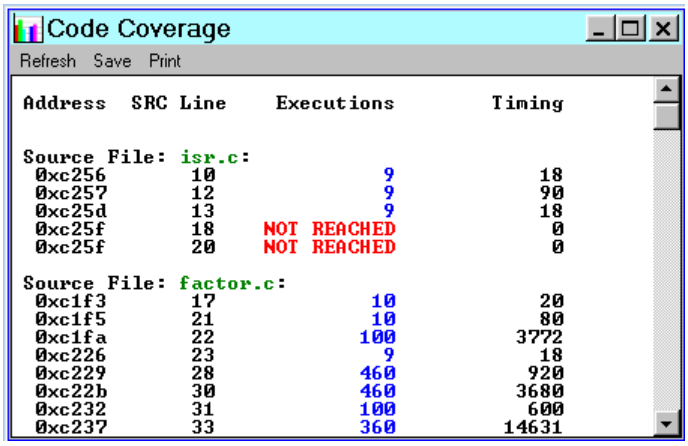
Chronograms

Displays a graphical function call time-line for your application's functions. The Chronology information complete with enter and exit cycle counts can be saved in a text report for future use.



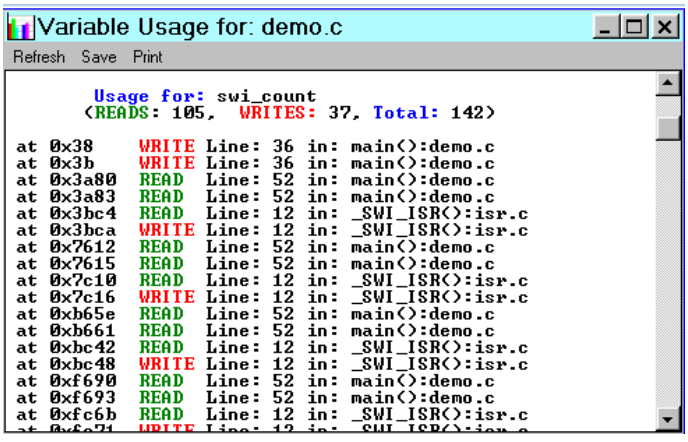
Code Coverage

ZAP SIM's Code coverage feature displays information on executed source lines and address information. Choose to show all coverage, only code executed or only code not executed. Optional text report are available. Code coverage information can also be displayed in each source browser window by selecting "show" from the title bar.



Variable Usage Reports

ZAP SIM also includes a variable usage report feature that details global variable usage for each variable. ZAP totals the number and type of access for each variable. It also details each access to include the address of the instruction accessing the variable, the type of access (read or write) and the line number in the source file that performed the access.

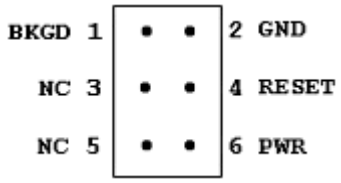


Background Debug Mode (BDM) Configuration

ZAP is available for Freescale's Background Debug Mode interface on the HCS08, HC12, HCS12, S12X/XGATE, 68HC16 and 68300 processors. This version of ZAP uses a PC USB or parallel port to interface directly to the target system using a BDM interface cable.

ZAP 6812 BDM and ZAP HCS08 BDM

ZAP 6812 and ZAP HCS08 supports P&E Microcomputer Systems' BDM Multilink (LPT and USB) and Cable12 via a 6 pin BDM target interface as shown below:



ZAP 6812 BDM currently supports all of the HC12, HCS12 and S12X/XGATE processors. ZAP HCS08 supports all members of the HCS08 family. These version of ZAP include all of the standard ZAP features and the following additional processor specific features:

FLASH and EEPROM programming - ZAP performs one step/one file downloads to FLASH, EEPROM and RAM.

Real-Time Debug – Debug code stored in on-chip FLASH or EEPROM. Popular for in-the-field debugging and reprogramming production systems.

Bank Switching – ZAP includes complete support for the on-chip bank switching mechanisms.

Real-Time C Trace – ZAP HCS08 BDM includes a real-time C and Assembly trace, complex triggers and profiling features using the HCS08's debug module.

Hardware Breakpoints – ZAP supports the on-chip hardware breakpoints for code and code and data breakpoints. Software breakpoints are also available when debugging out of RAM or in single step mode.

Real-time Memory – monitor and modify any data objects and memory while the processor is running.

Multiple Execution Modes – ZAP offers 3 execution modes for different target environments and chip modes.

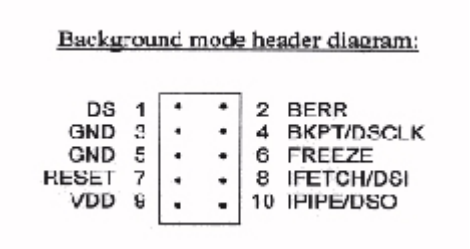
(1) Choose Hardware Breakpoint mode for real-time execution and debugging in single chip mode with program located in on-chip FLASH and/or EEPROM.

(2) Choose BGND mode for real-time execution and debugging with an unlimited number of software breakpoints. Code is executed in RAM.

(3) Choose "Single Step" for unlimited breakpoints when real-time execution is not required. ZAP uses instruction steps for all execution in this mode.

ZAP 6816/68300 BDM

ZAP for 68HC16 and 68300 uses P&E Microcomputer Systems' CABLE_16/32 and ICD16/32 and the following 10 pin BDM interface.



ZAP CABLE16/32 currently supports all of the HC16 and 683000 processors. This version includes all of the standard ZAP features and the following additional features:

Real-time Debug: C and assembly Source level debugging of code executed from RAM with an unlimited number of software breakpoints.

In-Circuit Emulator Configuration

The ICE configuration of ZAP is often used in the latter stages of development to fine tune optimizations and track down hard to find bugs in real-time embedded applications. The ZAP interface is currently available for the following emulators:

Freescale MMEVS/MMDS™ 68HC08

Freescale , MMEVS/MMDS™ 68HC05

This version provides additional features to fully support the Bus State Analyzer with C and Assembly source trace, complex event triggers and dual real-time memory support for viewing and updating data objects while the program is executing.

ZAP MON08

ZAP MON08 is designed to work with the 68HC08 MON08 serial interface and P&E MON08 Multilink cable. This version supports FLASH programming, the on-chip hardware breakpoint and security mechanism to provide a low cost real-time debugging solution directly on the target system.

ZAP for inDart design kits

ZAP for inDART is available for inDart-HC08 and inDart-HCS08 design kits. The HC08 version uses the on-chip MON08 module to provide real-time debugging, flash programming and security support. The HCS08 version uses the on-chip BDM and debug modules to provide additional features including real-time trace and triggers.

Supporting Embedded Innovation
Since 1983

freescale™
Alliance Member

For more Information please contact one of our corporate offices or visit our website:

Cosmic Software, Inc.
400 West Cummings Park, Suite 6000
Woburn, MA 01801-6512 USA
Phone: +1 781 932 2556 Fax: +1 781 932 2557
Email: sales@cosmic-us.com
web: www.cosmic-software.com

Cosmic Software France
33 Rue Le Corbusier, Europarc Creteil
94035 Creteil Cedex France
Phone: +33 1 43 99 53 90 Fax: +33 1 43 99 14 83
Email: mailto:sales@cosmic.fr
web: www.cosmic.fr

Cosmic Software UK
Oakwood House
Wield Road, Medstead
Alton, Hampshire
GU34 5NJ, U.K.
Phone: +44 1420 563498 Fax: +44 1420 561946
Email: *sales@cosmic.co.uk*

Cosmic Software GmbH
Rohrackerstr 68 D-70329 Stuttgart Germany
Tel.+49 711 420 4062 Fax +49 711 420 4068
Email: sales@cosmic-software.de
web: www.cosmic-software.de