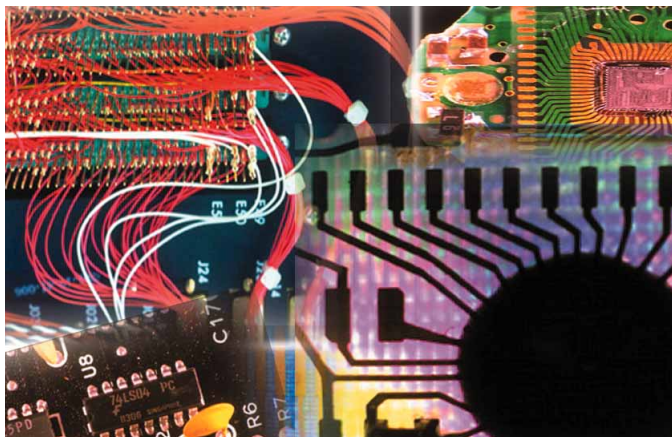


**MICROTEK**

In - Circuit Emulators

the
SWAT™ SoftWare
Analysis Tool

A Tool to Aid in Testing During the Development Cycle

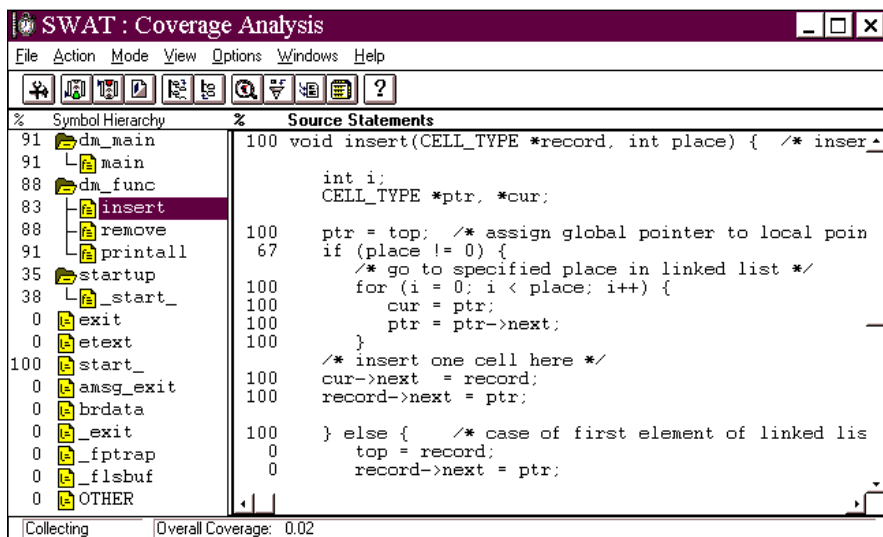
As designs are developed with more and more aggressive schedules it becomes more important that each portion of the design works as expected. The most common way to ensure this is through component testing prior to the final integration stage. The SWAT™ (SoftWare Analysis Tool) option for the Microtek PowerPack® line of emulators provides the developer with a tool to perform rigorous testing right from the start of the project. By running the tests and saving the results during the development, the designer is able to pass along not only his design, but a great deal of the testing required for final integration. The result is a much smoother transition from the designer's workbench through system integration and into the manufacturing environment.

A typical coding development begins with the engineer developing startup code for his project and then proceeding to the main portion of his program where functions are added to perform the application at hand. Once code has been written, it needs to be tested to ensure it operates as expected and it has the performance necessary to meet the system's needs. To aid in this process the PowerPack emulator with SWAT allows the user to perform code coverage during the development of functions and performance analysis during the system integration stage. These testing steps are easily integrated into the development cycle because the code requires no additional "instrumentation" to run the tests. The user needs only activate the test feature needed and the results are collected using the standard code with no additional compiling and linking required. Code coverage provides assurance the code "works as advertised" early in development. Performance analysis identifies bottlenecks early in development.



After running the suite of tests, the developer can then check the code coverage results from the SWAT and decide if more tests are necessary to exercise the portions of code that were reported not covered by the SWAT. If the test cases

are generated against a product requirements list, then the user works "backwards" by answering the question, "what requirement-level input value(s) must the product be exercised with, and what state does the software have to be put into to cause this segment of code to be executed?" These additional values and input sequences are added to the test suite. The measurement is run again to determine if the additions did their jobs. The user iterates through this process until satisfied that the code is being thoroughly tested.

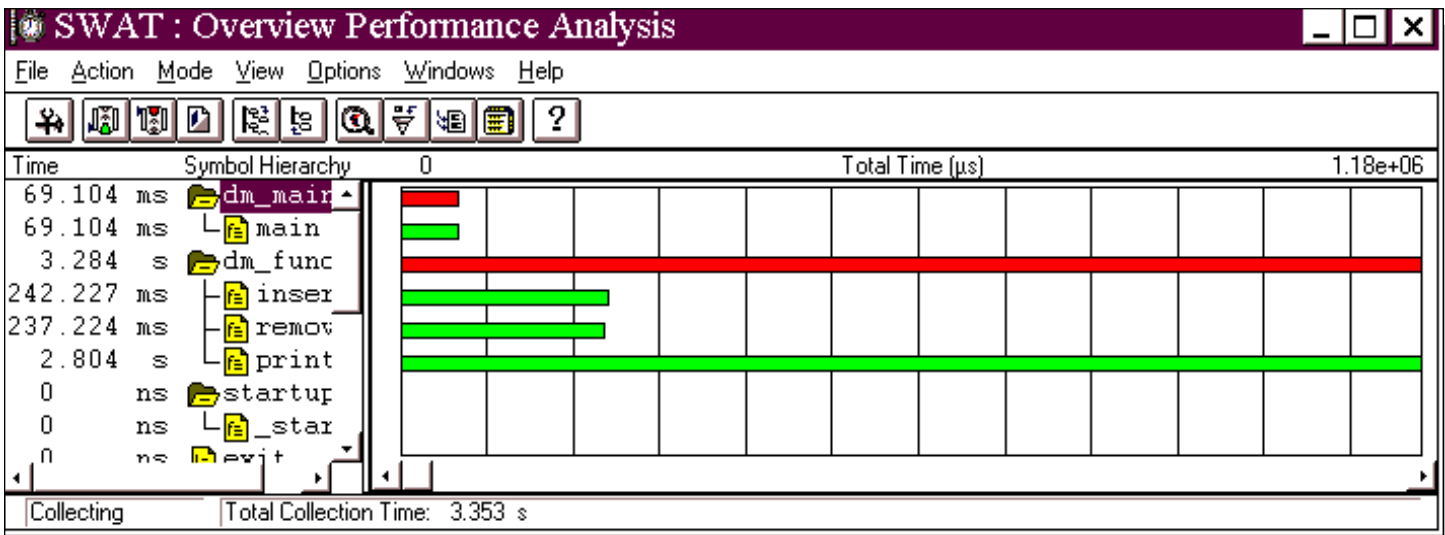
*continued on page 2*

continued from page 1

Once the user has developed a test suite that gives the developed code complete coverage, he can store these tests and the results for use after the product has been integrated with remaining code and hardware. These are called regression tests. The developer has checked each logical section of his code and has good confidence that the code "works as advertised". A designer would repeat these steps with each module developed. The resulting code has the benefit of having been completely executed during testing. A further benefit is the creation of regression tests during development.

The benefits of this approach multiply as you approach system integration. The resulting test suites are available to be used as a basis for regression testing, easing the cost and time necessary to build complete integration tests suites. These tests should then be augmented with more application specific tests to insure the functionality provided by the testing routines integrate and play correctly together.

At the system integration stage of development two testing suites need to be used. One test suite is the regression tests. The second is performance tests. This stage of testing covers any unexpected performance issues. Perhaps your application has a feature that uses functions that appeared quick enough stand alone, but are slow when used together. These issues are then addressed by performance testing. This approach leads to a product that has reliability built in rather than hammered on in the last testing phase. Performance and quality have been addressed from the start of the project using a development tool that provides support for testing the overall design. Time spent preparing the code for incremental testing is minimal, since no code instrumentation is necessary. No additional compiling and linking are necessary.



System Features at a Glance:

Coverage Analysis

- 100% accurate instruction execution coverage even for cached and queued processors.
- Can provide coverage analysis over two 1 Megabyte ranges with byte level resolution for 386 processors. The 486 class processors have two 512k ranges with byte level resolution. Options exist to double the amount of code that can be covered.
- Allows the user to view coverage to the object level.
- Mixed mode display automatically identifies unexecuted code.

Overview Performance Analysis

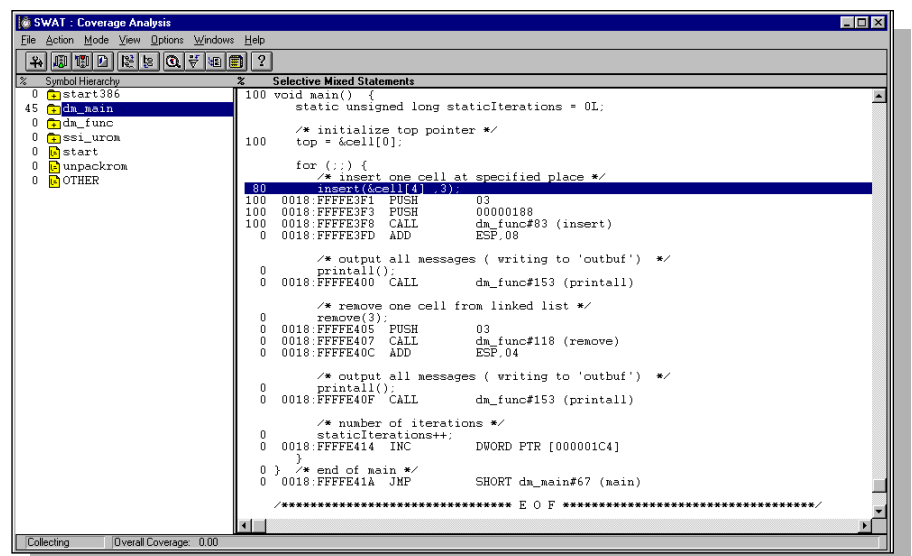
- The SWAT hardware accumulates time of execution at each program address and provides summations of time for symbolic ranges of modules and functions.
- The 1 megabyte hardware option can cover 256K bytes of address space, with two definable map ranges of 128K each. Resolution of coverage can be changed to provide 512K at 2 byte granularity, 1M at 4 bytes, up to 64M at 256 byte resolution.
- Can work in memory areas of 512k with resolution of 2 bytes.
- Uses an independent clock source to provide 30ns time resolution.
- Employing the symbol table, the SWAT provides accumulation results for modules, functions, and line numbers (386EX only).
- Measures time outside of defined ranges.
- Provides histogram views of absolute accumulated time per range or percentage of total time, data as histograms by module or function; reports percentage of total timer, or absolute time spent in module or function.
- Results can be sorted by absolute or relative time, providing a quick view of program hot spots at the module or function level.

SWAT Source Level Interface

The SWAT performance tools are controlled by the PowerPack-SLD interface, which benefits from over 30 man-years of development effort. This full featured windows-based point and click interface provides complete C or ASM source level debugging.

PowerPack-SWAT interface feature list:

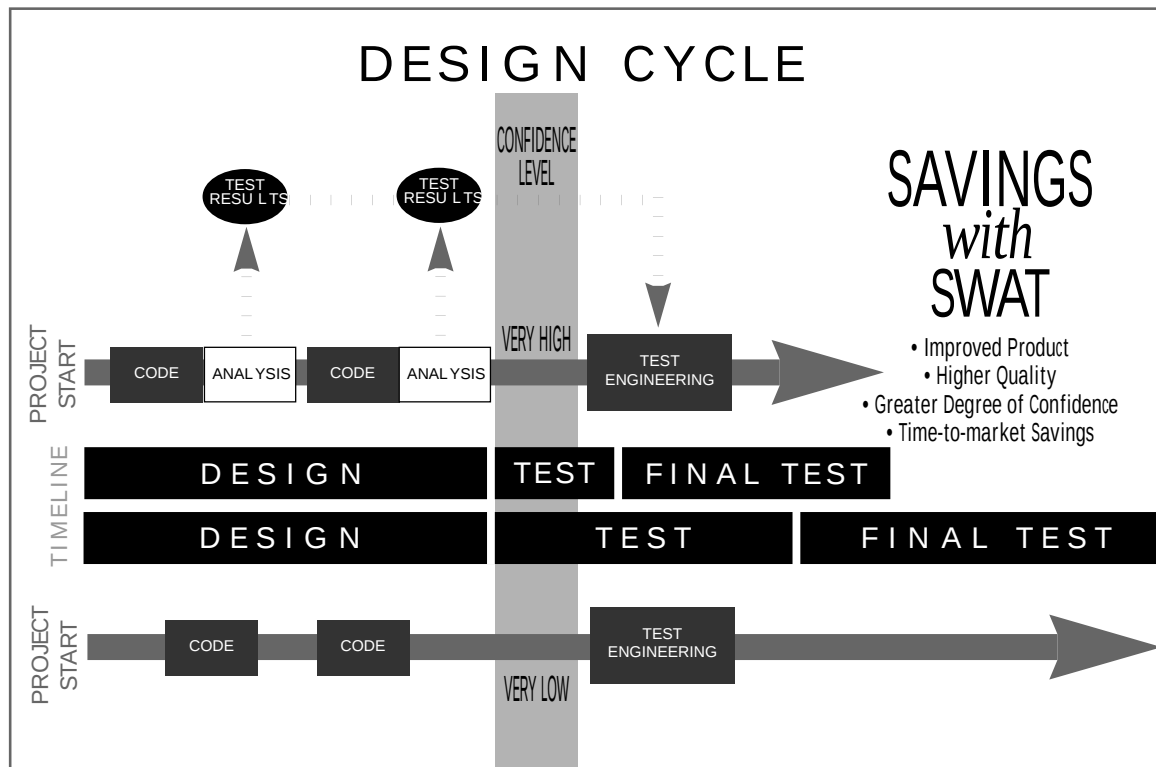
- Fully integrated into the PowerPack-SLD interface allowing the user to identify performance bottlenecks with SWAT and then use the emulator to debug the code.
- Histogram output showing absolute or relative time spent in modules or functions.
- Can select view of output to be sorted by results, by name or by address.
- Uses color annotation of source code to show what lines are covered/not covered.
- Allows the user to remove symbolic ranges from view and results calculation.
- Allows code coverage to be viewed at the assembly level providing the detail necessary to uncover all unexecuted instructions.



Everyone knows debugging is easier in small isolated pieces of code than it is in large integrated systems.

This is a basic concept of object oriented programming. Using tested objects in new projects reduces time to market by eliminating delays in system integration.

The Microtek SWAT product assists the engineer in testing his new routines thereby speeding time to market by reducing delays in system integration.



PRODUCT	CODE	DESCRIPTION
386SWAT with 1 MEG	PP-SWAT386-1MEG	386 Software Performance Analysis Board with 1 Megabyte
386SWAT with 4 MEG	PP-SWAT386-4MEG	386 Software Performance Analysis Board with 4 Megabytes
486SWAT with 1 MEG	PP-SWAT486-1MEG	486 Software Performance Analysis Board with 1 Megabyte
486SWAT with 4 MEG	PP-SWAT386-4MEG	486 Software Performance Analysis Board with 4 Megabytes



Microtek makes it easy to keep your emulator up to date and in good working condition, which is critical to our mutual success. Join our Gold Support program and enjoy all the benefits of a complete warranty.

For more information about Service Support Options to match your project requirements, contact Lisa Rice at (800)886-7333x401 or (503)533-4463x4038.

FAX



(503) 533-0956

E-MAIL



info@microtekintl.com

WEB SITE

<http://www.microtekintl.com>