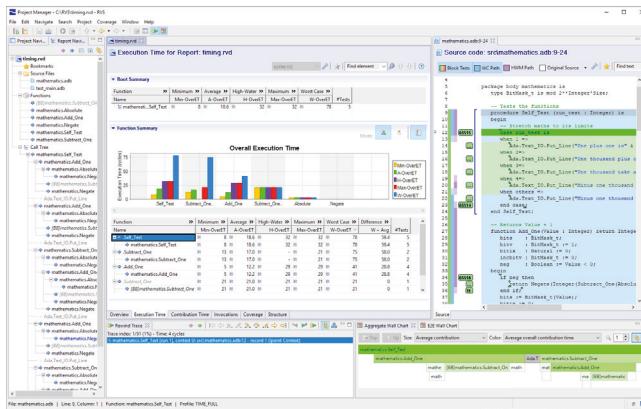**RAPITA** Systems
A DANLAW Company

Measurement-based timing and WCET analysis with RapiTime

Rapi**Time**

How can RapiTime help you?

Rapi**Time** is an advanced tool for performing timing and worst-case execution time (WCET) analysis, designed specifically to work with embedded targets and to satisfy certification requirements. The timing metrics produced by Rapi**Time** can be used to demonstrate that you have satisfied DO-178B/C objectives.

Using a unique hybrid static/dynamic analysis approach, Rapi**Time** automates timing analysis on embedded systems to provide detailed information on their timing behavior and to help identify timing issues and optimize code for timing behavior.

Benefits of using RapiTime

Rapi**Time** helps you reduce the cost, time and effort you need to perform timing analysis and optimization on even the most complex and demanding critical real-time embedded systems.

You can use Rapi**Time** to automate timing data collection and analysis even on very large systems. In one case study, Rapi**Time** produced timing behavior that a customer took 8 months to collect manually in just one day.

Rapi**Time**'s minimal overhead means that you can perform timing analysis in every test run, making timing information available throughout your software development. This information will help you to identify timing issues early in development and minimize your code's WCET.

RapiTime use cases

- Demonstrate that software executes within its time constraints.

- Understand timing behaviour when upgrading to new targets, even multi-core processors.

- Optimize code to upgrade legacy systems.

- Conduct WCET/high water mark analysis.

- Address DO-178B/C guidelines.

How does RapiTime work?

Rapi**Time** performs static analysis of code and instruments it automatically. When you run your code on-target, Rapi**Time** collects a trace of the program execution that includes timing data. Rapi**Time** then processes this trace to produce qualifiable reports of the timing behavior of your code that you can view using the Rapi**Time** GUI.

The timing reports that Rapi**Time** produces, along with its trace rewind feature, let you quickly identify where your optimization effort will provide the greatest improvements to timing behavior.

Rapi**Time**'s integration and instrumentation is flexible, and its extremely low instrumentation overheads ensure that it can efficiently analyze the timing behavior of even the most complex critical embedded software.
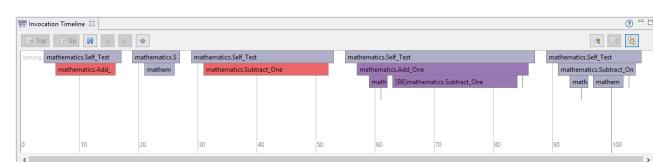
Key features of RapiTime

Timing analysis

- Automated collection of timing metrics on-target and on-host
- Powerful hybrid WCET analysis engine
- High water mark (HWM) analysis
- Analysis configurable to include or exclude specified modules/functions/directories
- Time band analysis to automatically configure instrumentation level
- Calculation of detailed timing metrics for each function and sub-function:
    - Minimum, maximum and average execution time
    - Execution time density
    - Contribution to worst-case and HWM paths
- Collect and analyze metrics from performance monitoring counters on hardware

Analysis engine

- Context-sensitive analysis
- Support for function pointers and recursion
- Powerful annotation mechanism
- Complex code structures

Language support

- Ada 83, 95, 2005 and 2012, compilers including GNAT Pro and Green Hills
- C and C++, compilers including Visual Studio, GCC, Diab and TASKING
- Assembly code insertions
- Mixed language source code

Build integration

- Multiple strategies available:
    - Compiler wrappers
    - Clone integration
    - Scripting into build system directly
- Support for very large code bases
- Support for legacy compilers
- Instrumentation can be split between build cycles
- Shared integration with other R**VS** tools

Target integration

- Flexible trace collection using CAN, Serial, Ethernet, debuggers, in-memory trace buffers, hardware I/O tracing, hardware tracing support e.g. Nexus, and our own **RTB**x data logger
- Extremely low overhead instrumentation library for 8, 16, 32 and 64 bit architectures
- No library/run-time dependencies or dynamic memory requirements

- Support for zero overhead instrumentation on selected targets
- Timing analysis across power cycles (subject to hardware requirements)
- Data collection freeze and reset to eliminate accidental tracing
- Extremely fast, lock-free, thread-safe tracing mechanism
- Support for multitasking and multi-core processors

Tool qualification

- Qualification kit and service to support DO-178B/C tool qualification

Third party integration

- Tools such as Mx-Suite™, MATLAB Simulink and GNAT GPS
- Continuous build servers e.g. Jenkins, Bamboo
- Debuggers e.g. Lauterbach, i-SYSTEM

Integrated testing environment

- Summary and detailed results views
- Invocation timeline, aggregate profile and treemap charts to help understand timing behavior at a glance
- Trace rewind feature to debug timing behavior
- Code viewer:
    - View source code alongside pre-processed and instrumented code
    - Color-coded by WCET and high water mark paths
- Show other code metrics e.g. #LOC, #loops
- Aggregate timing metrics by directory, file and functions
- Multiple export formats: text, XML, CSV, image formats
- Merge results from different test runs, builds and strategies
- Compare reports
- Database-like search function

Compatibility

- Runs on host operating systems
    - Windows 7+ and Windows Server 2008 R2+
    - Linux distributions including Ubuntu and Red Hat
- Results can be collected from systems without supported operating systems and transferred to a supported system for analysis

Licensing

- Enterprise license gives you access to new versions, support and maintenance
- One-year support and maintenance included in purchase price
- Single price for all features
- Licenses transferrable across projects