



 **RAPITA** Systems
A **DANLAW** Company

Multicore timing analysis solution *for automotive*

- » Ensure multicore code is free from interference
- » Evaluate multicore hardware
- » Optimize multicore code for timing performance

Multicore timing analysis

We provide a unique solution to support the use of multicore hardware in critical systems. This helps you ensure that your multicore code is free from interference for ISO 26262 compliance, reducing migration risks and opening up the benefits of increased performance available from using multicore hardware.

“*Multicore systems are becoming more popular in critical embedded system development due to the increased performance they offer.*

Our multicore timing analysis solutions solve an important challenge in using these complex systems; ensuring that the software is free from interference, meets timing deadlines and can satisfy certification standards.

”

Dr. Guillem Bernat, CEO of Rapita Systems

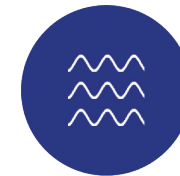
A unique solution

Multicore systems are being adopted rapidly in the critical automotive software industry to provide additional performance for complex systems such as autonomous driving and ADAS. New methods are needed to analyze the timing behavior of these systems in line with ISO 26262 standards.

Combining expert knowledge from dedicated engineers, products from groundbreaking academic research and industry-leading software tool support, our solution to multicore timing analysis is truly unique.

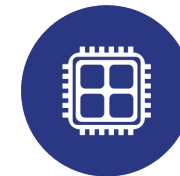
Use cases

Our solution supports a variety of use cases when migrating to, using and verifying multicore systems:



Ensure freedom from interference

Produce timing evidence for multicore systems to ensure that your system is free from interference caused by timing effects and complies with ISO 26262 standards.



Evaluate multicore hardware

Evaluate advanced complex multicore platforms including GPU platforms against performance criteria, taking into account the effects of interference from simultaneous use of shared resources.



Optimize code for timing

Optimize multicore code for execution time behavior, ensuring it is free from interference caused by timing effects and meets timing deadlines.

Benefits of our approach

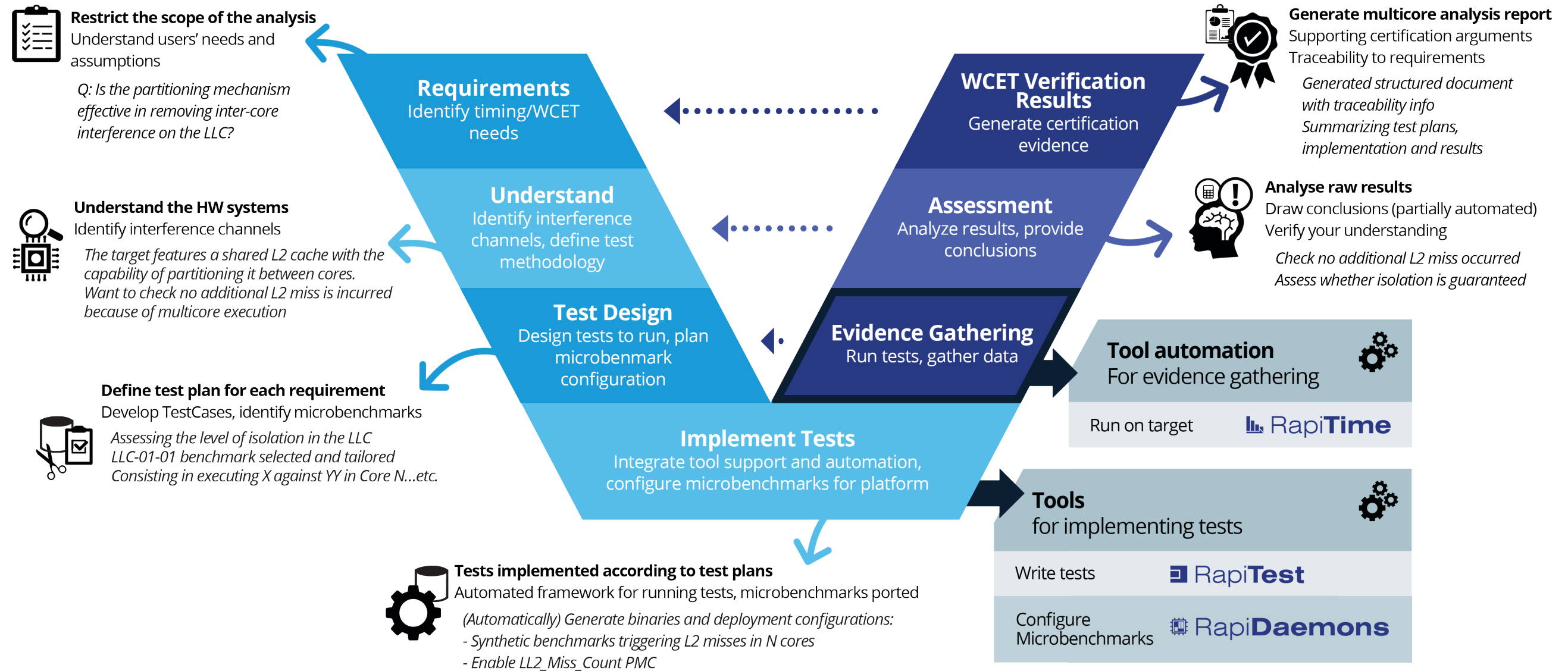
Our approach not only identifies interference channels in multicore systems, but also quantifies them and takes them into account during timing analysis. We take advantage of industry-leading tool automation support to provide a cost-effective solution to analyze multicore timing behavior and ensure that multicore systems are free from interference and thus comply with ISO 26262 standards.

Working with us

- We recognize that every project is different, and work with you to meet your needs.
- We run services at our engineering facilities in the UK or US.
- We can answer multicore timing questions and produce evidence for you, or implement a method and provide training so you can do so yourself.



How it works



Interference, and freedom from it

Multicore code can suffer *cascading failures* as, due to heavy use of a shared resource, code running on one core can affect the execution time of code running on another core, causing that code to fail to execute as expected. This is defined as *interference*, and, according to ISO 26262 standards, automotive code should be free from this interference.

Our multicore timing analysis solution helps you verify that your code is free from interference and does not suffer from cascading failures due to simultaneous use of shared resources.

Microbenchmarks

To examine the effects of *interference* on multicore timing behavior, our multicore timing services use microbenchmarks.

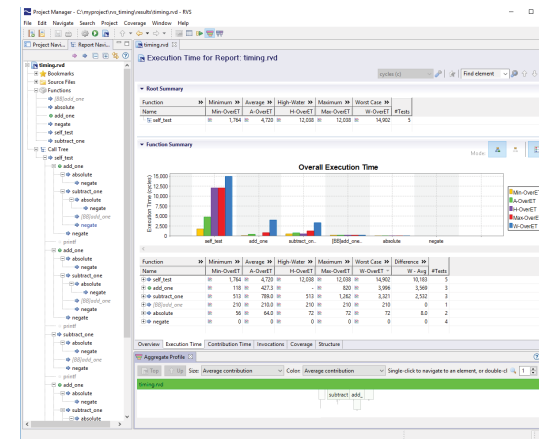
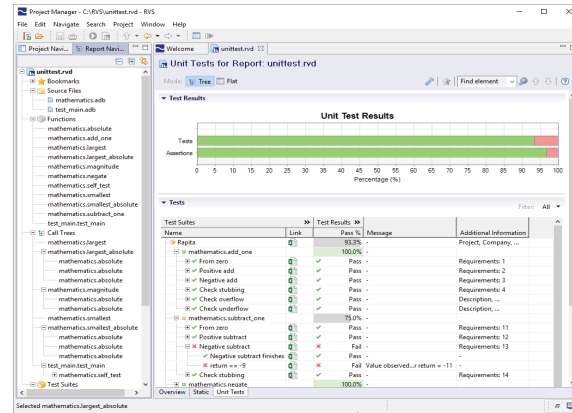
These are specially designed applications that can be integrated with the system under analysis to create a configurable degree of interference from contention for shared resources such as caches and buses when running tests.

Tool support

RapiTest

RapiTest helps to produce and run tests that exercise multicore software for execution time behavior while taking into account the effects of resource contention and interference (through applying **microbenchmarks**).

RapiTest automatically converts tests into a test harness that can be run on the multicore hardware.



RapiTime

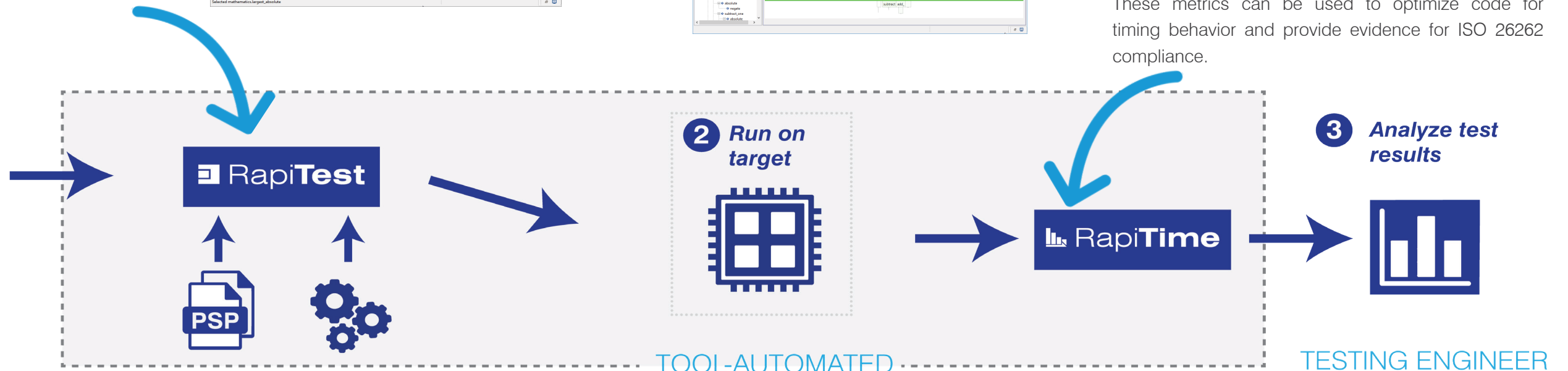
RapiTime automatically calculates execution time metrics when multicore software runs on its target hardware, and reports them in a format that is easy to understand.

These metrics can be used to optimize code for timing behavior and provide evidence for ISO 26262 compliance.

1 Write tests



TESTING ENGINEER



RapiDaemons

RapiDaemons create resource contention while analyzing a multicore task under analysis. Some microbenchmarks are generic and are available as a standard library, while some are platform-specific and must be adapted to the platform under analysis through an integration service (right).

Integration

Engineering Service

To perform multicore timing analysis, we integrate our tools into your multicore system. This involves developing a **Platform Support Package (PSP)** defining configurations of your system, configuring **microbenchmarks** to generate interference on your system, and integrating RapiTest and RapiTime to work with your development environment.

During the integration, one of our *Field Application Engineers* will work with you either remotely or on-site to set up the integration, and will produce a report describing how it works.

*Meeting global testing needs in the critical
embedded software industry since 2004*

 Rapita location  Distributor  Customer



Get in touch

Each safety-critical project is different.
Contact us to arrange a custom solution that meets your needs:

Visit: www.rapitasystems.com/contact

Email: enquiries@rapitasystems.com

UK office
Tel: +44 1904 413945

USA office
Tel: +1 248-957-9801

Rapita Systems Ltd.
Atlas House
Osbalwick Link Road
York, YO10 3JB
UK

Rapita System Inc.
41131 Vincenti Ct.
Novi
MI 48375
USA