RAPITA SYSTEMS
A DANLAW COMPANY

Safety through quality

BROCHURE

Multicore Timing Analysis for Automotive

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 ground breaking academic research and industry-leading software tool support, our solution to multicore timing analysis is truly unique.

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.

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.

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

Restrict scope of the analysis

Q: Is the partitioning mechanism effective in removing inter-core interference on the LLC?

Understand the platform

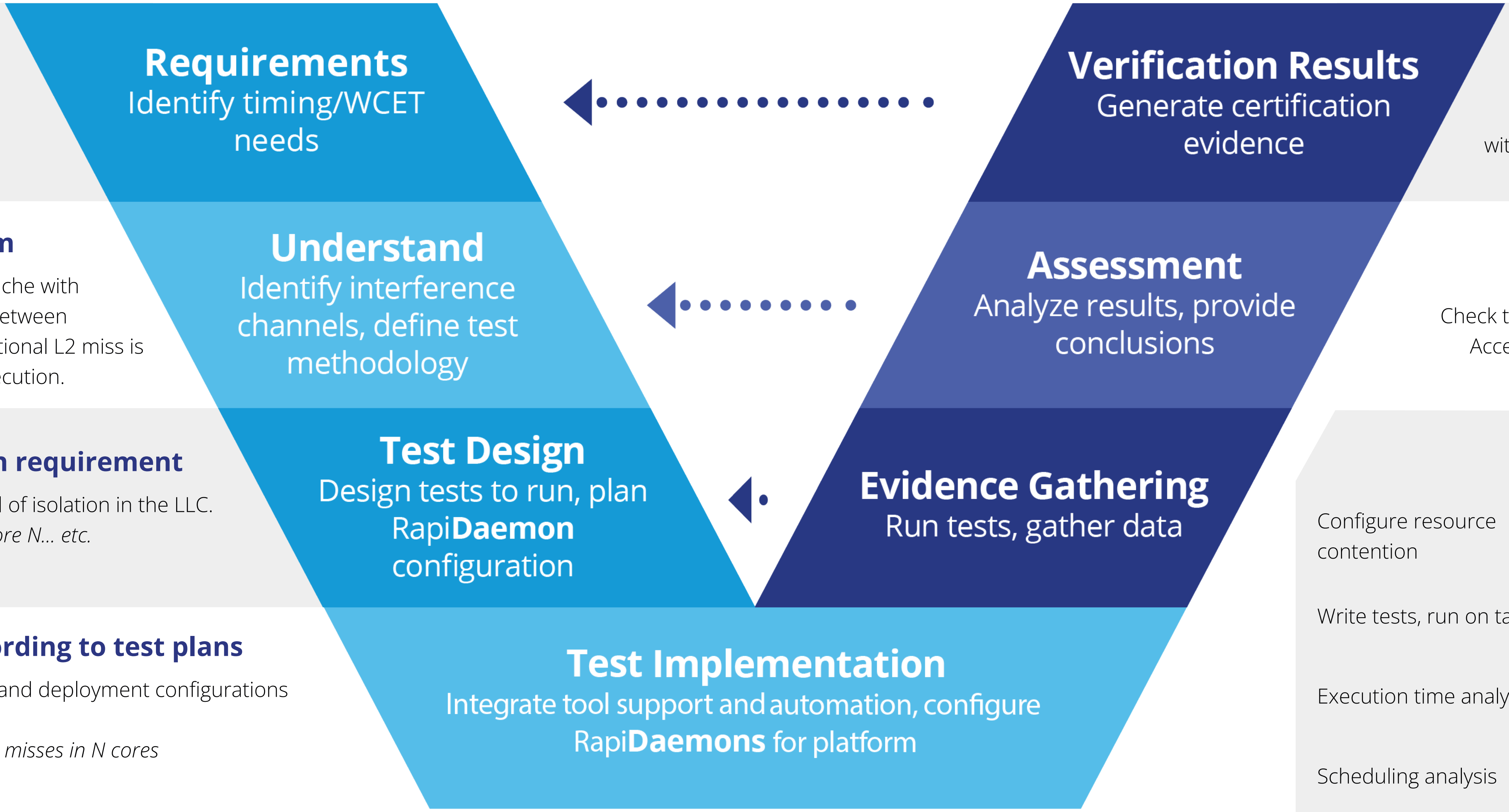
The target features a shared L2 cache with the capability of a partitioning in between cores. Need to check that no additional L2 miss is incurred because of multicore execution.

Define test plan for each requirement

Tests designed to access the Level of isolation in the LLC.
Test shall execute X against YY in Core N... etc.
*LLC-01-01 Rapi***Daemon** *selected.*

Tests implemented according to test plans

(Automatically) Generate binaries and deployment configurations e.g:
- *Synthetic contenders triggering L2 misses in N cores*
- *Enabling LL2_Miss_Count PMC*



Requirements
Identify timing/WCET needs

Understand
Identify interference channels, define test methodology

Test Design
Design tests to run, plan Rapi**Daemon** configuration

Test Implementation
Integrate tool support and automation, configure Rapi**Daemons** for platform

Evidence Gathering
Run tests, gather data

Assessment
Analyze results, provide conclusions

Verification Results
Generate certification evidence

Generate multicore analysis report

Automatically generated document with traceability info summarizing test plans, implementation and results.

Analyze raw results

Check that no additional L2 miss occurred. Access whether isolation is guaranteed.

Tool Support

Configure resource contention	Rapi**Daemons**
Write tests, run on target	Rapi**Test**
Execution time analysis	**RTB**x
Scheduling analysis	Rapi**Task**

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.

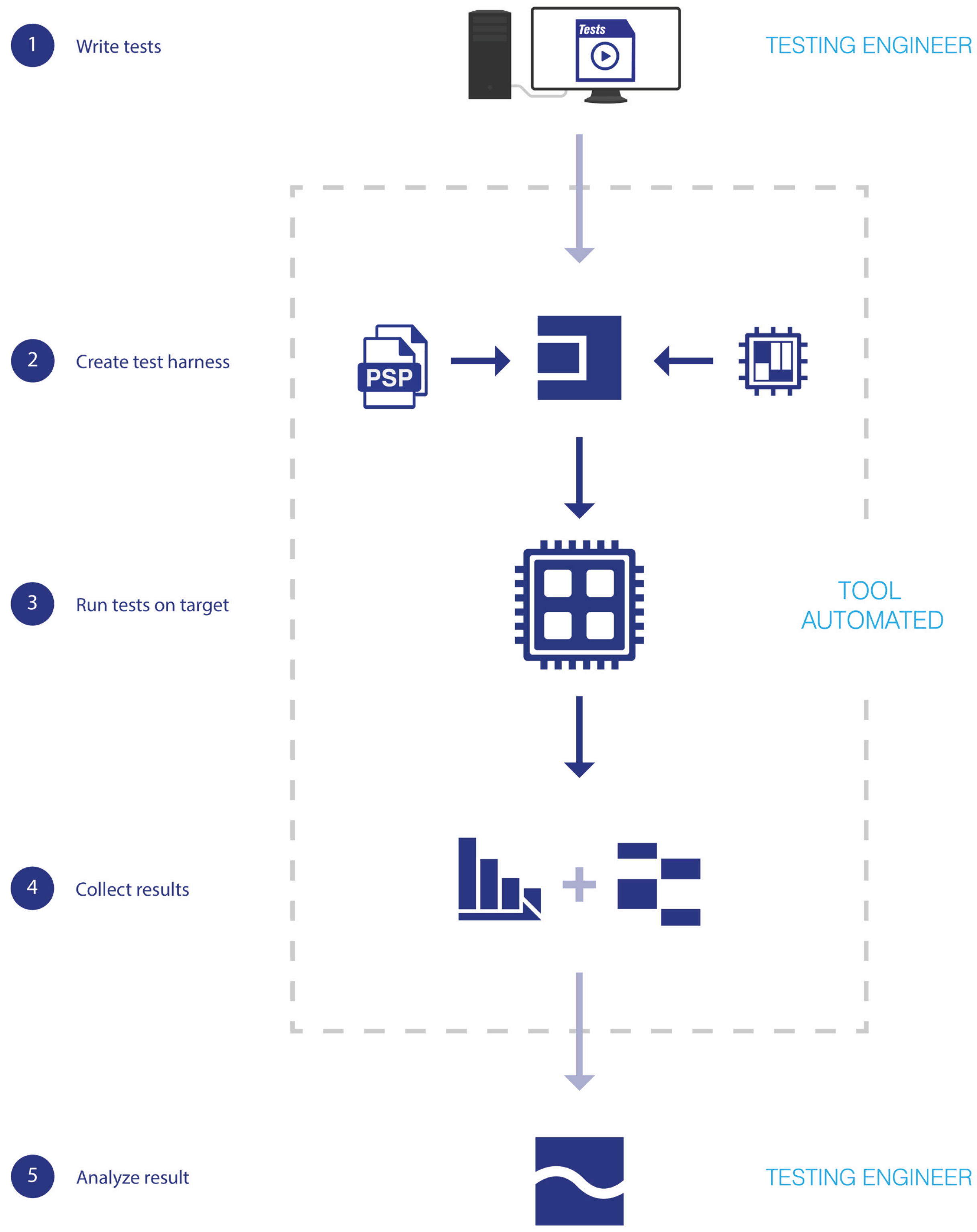
Our multicore timing analysis solution helps you verify that your code is free from interference and does not suffer from cascading failures due to simultaneous use of shared resources.

Working with us

To examine the effects of resource contention and interference on multicore timing behavior, our multicore timing services use Rapi**Daemons**.

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

Tool support



1 Write tests — TESTING ENGINEER

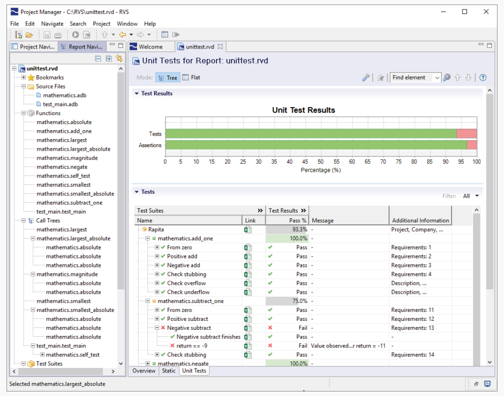
2 Create test harness

3 Run tests on target — TOOL AUTOMATED

4 Collect results

5 Analyze result — TESTING ENGINEER

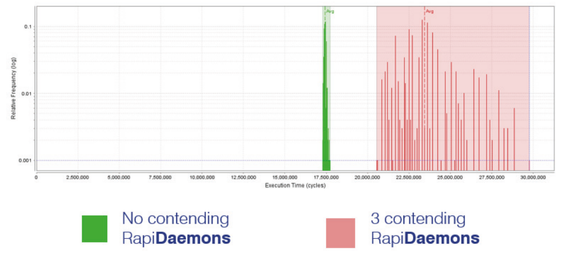
Rapi**Test**



Rapi**Test** 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 Rapi**Daemons**).

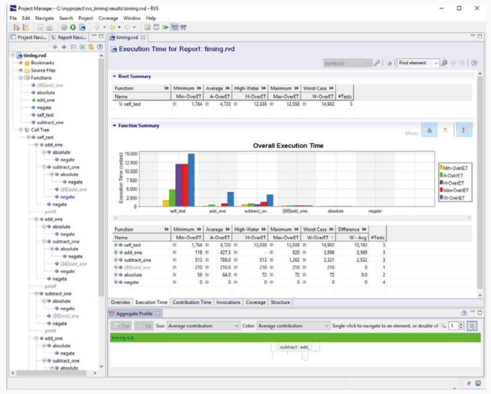
Rapi**Test** automatically converts tests into a test harness that can be run on the multicore hardware.

Rapi**Daemons**



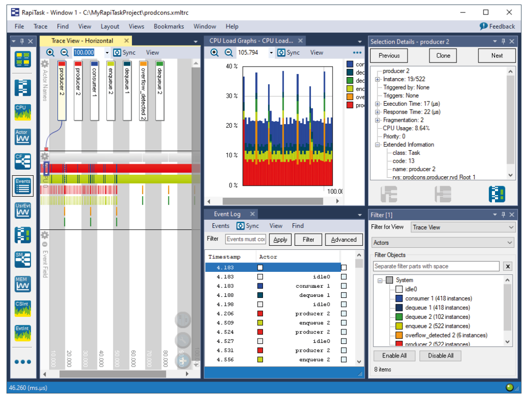
Rapi**Daemons** create resource contention while analyzing a multicore task under analysis. Some microbenchmarks are generic and are available as a standard library, while some are platform-specific and must be adapted to the platform under analysis through an integration service.

Rapi**Time**



Rapi**Time** 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.



Rapi**Task**

Rapi**Task** automatically measures and reports scheduling metrics for each task under analysis when multicore software runs on its target hardware.

These metrics can be used to identify system capacity issues and rare events such as race conditions in the software.

About Rapita

Rapita Systems provides on-target software verification tools and services globally to the embedded aerospace and automotive electronics industries.

Our solutions help to increase software quality, deliver evidence to meet safety and certification objectives and reduce costs.

Find out more

A range of free high-quality materials are available at:
rapitasystems.com/downloads

SUPPORTING CUSTOMERS WITH:

Tools

Rapita **Verification Suite**:

Rapi**Test**

Rapi**Cover**

Rapi**Time**

Rapi**Task**

Engineering Services

V&V Services

Integration Services

Qualification

SW/HW Engineering

Compiler Verification

Multicore verification

MACH^{178}

Multicore Timing Solution

Contact

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

+44 (0)1904 413945

Rapita Systems, Inc.
41131 Vincenti Ct.
Novi, Mi, 48375
USA

+1 248-957-9801

Rapita Systems S.L.
Parc UPC, Edificio K2M
c/ Jordi Girona, 1-3
Barcelona 08034
Spain

+34 93 351 02 05

rapitasystems.com

linkedin.com/company/rapita-systems

info@rapitasystems.com