



Industrial Application of Measurement Based WCET

Stephen Law

©2015 Rolls-Royce Controls and Data Services Inc.

The information in this document is the property of Rolls-Royce Controls and Data Services Inc. and may not be communicated to a third party or used for any purpose other than that for which it is supplied, without the express written consent of Rolls-Royce Controls and Data Services Inc.

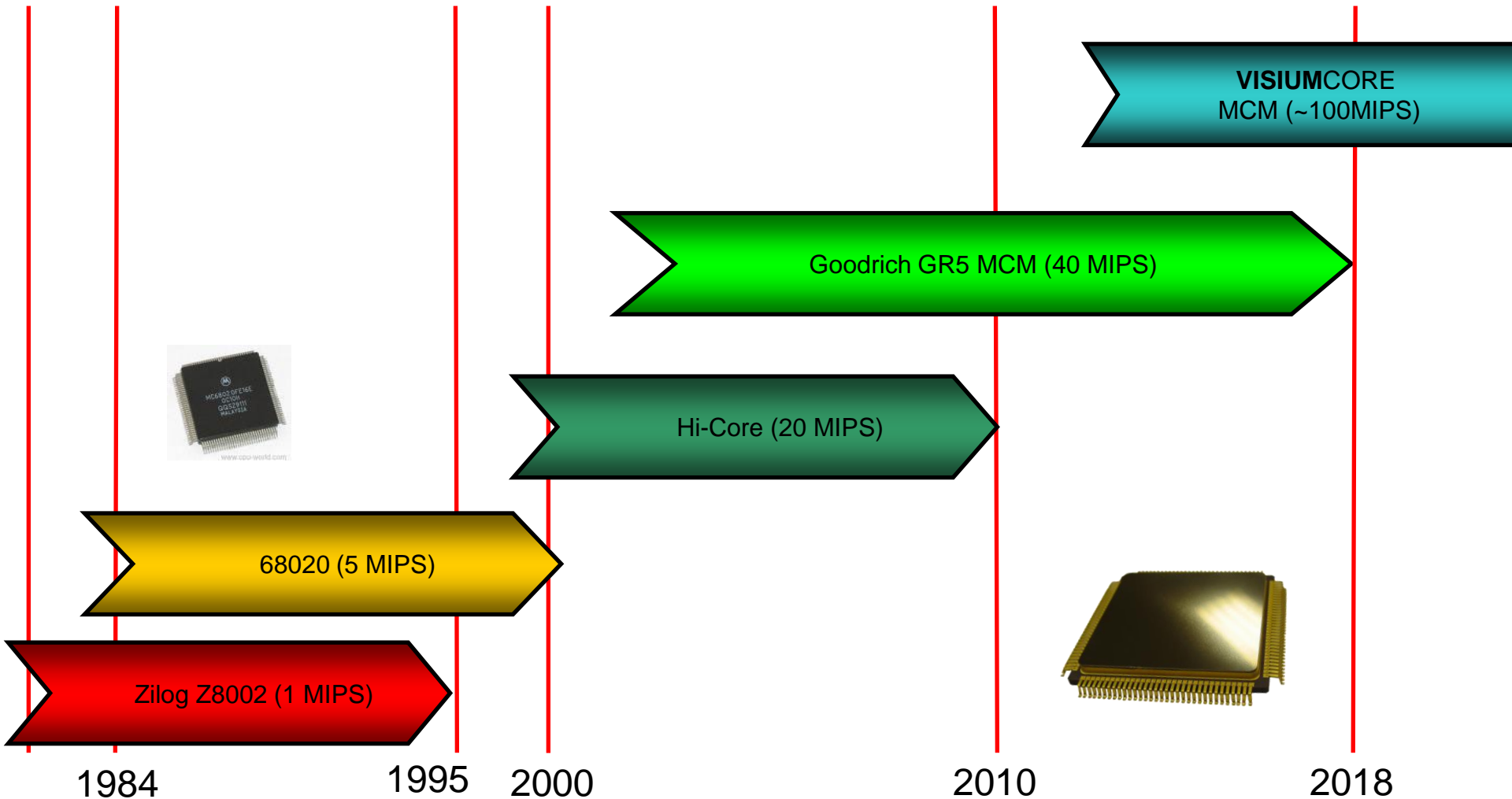
Introduction

- CDS develops aircraft engine control systems, predominantly for Rolls-Royce
- The software, **written exclusively in SPARK-Ada**, is developed according to DO-178B & C guidelines, DAL A
- Worst Case Execution Time (WCET) Analysis is a key verification activity for real-time systems.
 - Static Analysis
 - Measurement Based Analysis



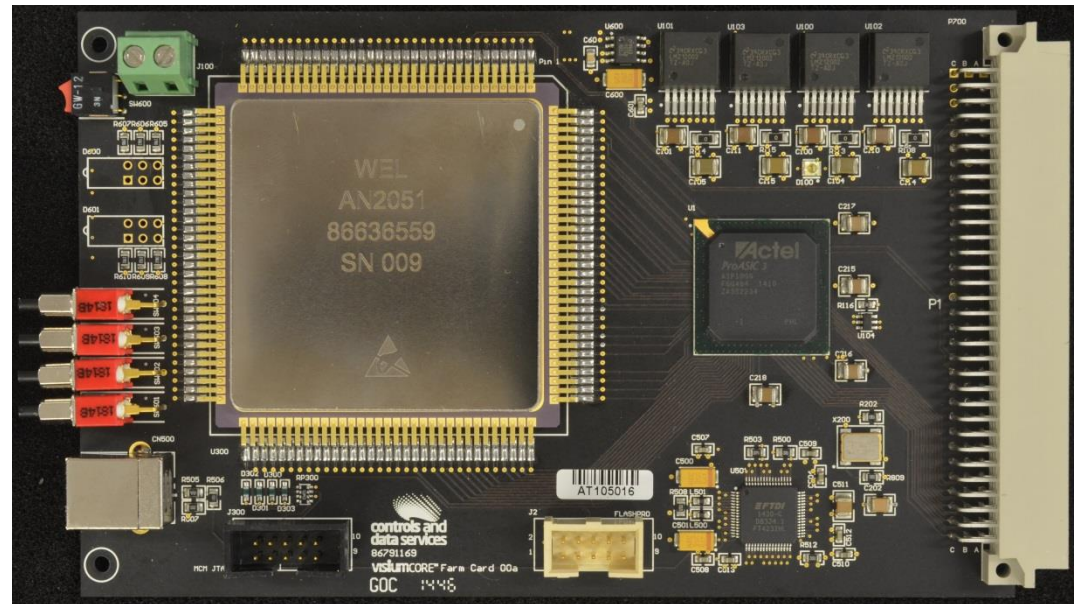
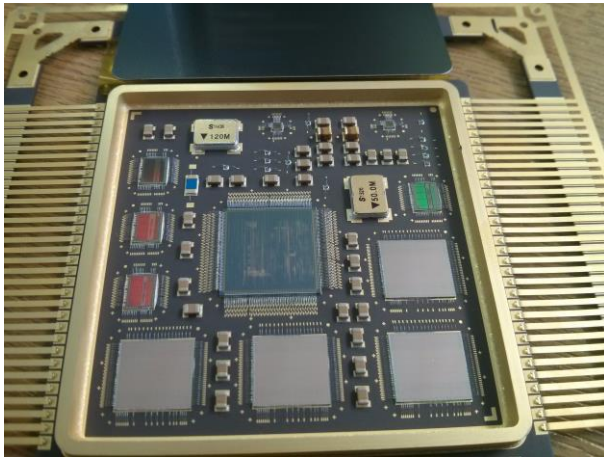
Our History

MIPS = Million Instructions per Second



The VISIUMCORE

- The **VISIUMCORE**, the latest iteration of the CDS processor, is designed specifically for software development in a DO-178C/DO-254 context



The **VISIUMCORE**

- The **VISIUMCORE**, the latest iteration of the CDS processor, is designed specifically for software development in a DO-178C/DO-254 context
 - Five-stage superscalar architecture featuring managed out of order execution
 - Time invariant instruction execution
 - Static branch prediction logic
 - No instruction or data cache
 - Independent instruction/data memory access/control
 - In-built trace output logic, provided through an independent execution unit and special tracing instructions
 - Each RapiTime instrumentation overhead <1 cycle.
 - Obsolescence avoidance – support for at least the next 20 years
 - Extensive SEU protection, detection and correction.

Industrial WCET Key Challenges

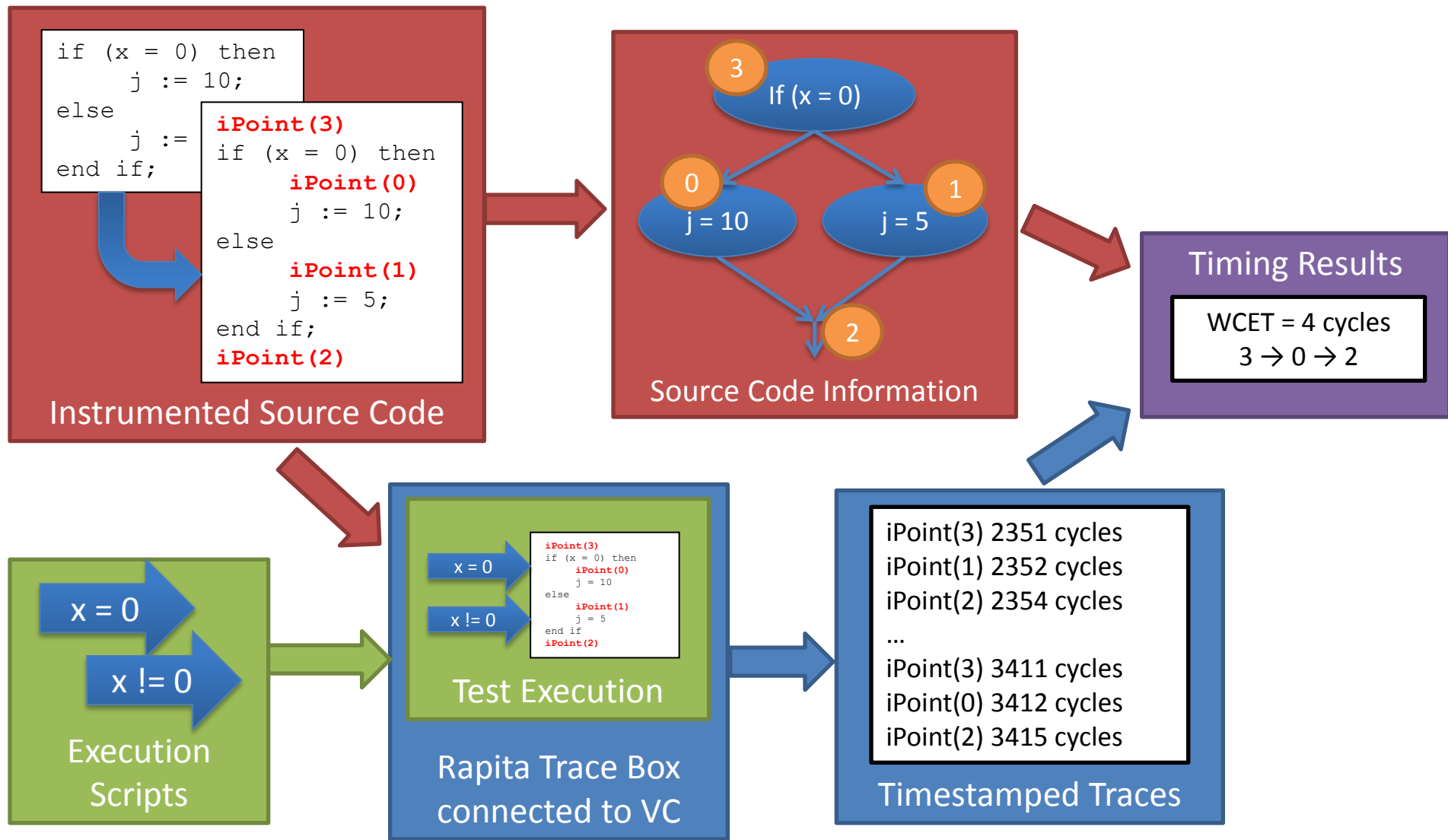
- Accurate WCET results, not optimistic, not overly pessimistic
 - A robust and reliable method, that meets CAA/EASA/FAA requirements
- Integrates into existing test strategy, build process, and able to analyse legacy software
- Scalable to a substantial project*
- Able to analyse at the component level
- Optimised code

**What scale are we looking at?*

A typical project consists of approximately 4000 individual component test scripts, which have been written over a 10 year timeframe

CDS Chosen Approach

- Software targeting the **VISIUMCORE** will be analysed using RapiTime, developed by Rapita Systems Ltd
- The solution offered easy integration with our processor, and our existing processes and software
- The tool is integrated with our existing low level testing (LLT) infrastructure
 - Completely automatic execution, controlled by existing LLT scripts
 - This allows testing to be performed on a single processor, long before full target hardware is available
- Minimal changes required to existing components:
 - Less than 1% of code items had to be altered
 - Less than 5% of test scripts required updates

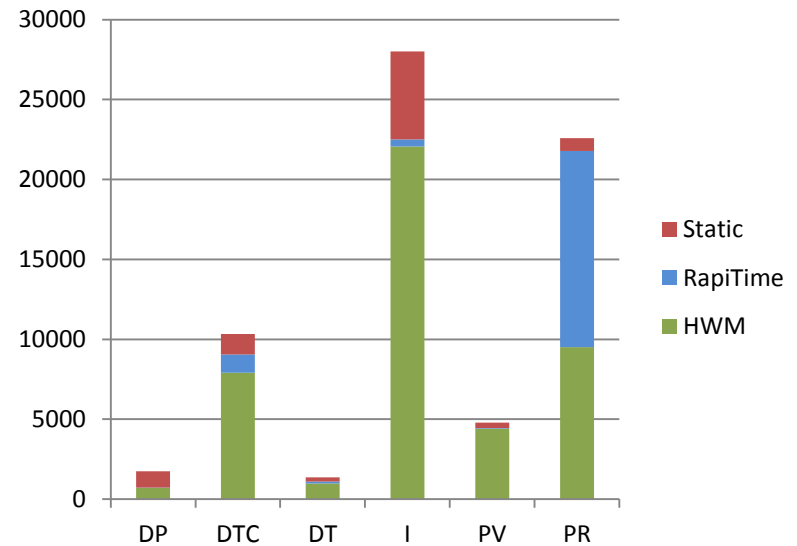
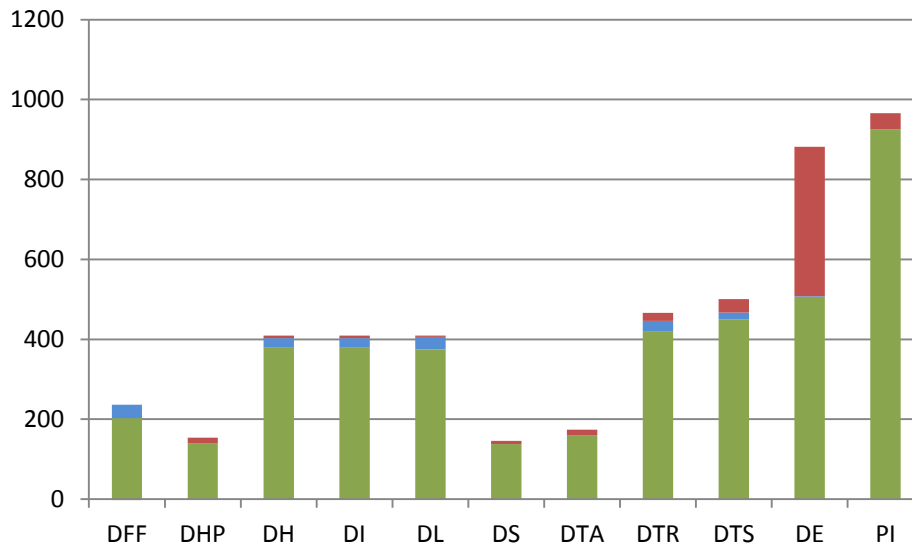


Argument for Suitability and Correctness

- *A software architecture amenable to analysis*
- *Extensive coverage provided through low level component tests*
- *A processor architecture designed for analysis*
- *Comparison between code under test, and code in the final system*
- *Combining unit tests to provide a system level result*

Initial Results

- An initial study tested the tool against an in house static analysis tool, targeting a previous version of the CDS processor



- Following the initial study the tool was trialled on the **VISIUMCORE** processor targeting two CDS engine controller projects
 - 8000 tests, all compared to HWMs

Conclusion

- RapiTime has been integrated as part of the CDS build and test process with minimal change required
- First adopting project now using the tool to generate certification artefacts
- The technology has been validated through a extensive study that has demonstrated acceptable WCET results and a viable process in a representative environment
- Future work:
 - Automatic software execution, aiming to reduce the tool's reliance on test scripts
 - Working with Rapita investigating ways to reduce the overhead introduced by instrumentation

Questions ... and Our History

MIPS = Million Instructions per Second

