WHITEPAPER

Unleashing the power of AURIX™ in performance benchmarks

TriCore™ Dhrystone and CoreMark results

Edition: v1.0
Date: 01/2024

www.infineon.com

Table of contents

Abstract

In this white paper, we will take a closer look at Dhrystone and CoreMark benchmarks and execute benchmark tests on Infineon's TriCore™ architecture for the AURIX™ TC3x and TC4x device families. It also outlines the benchmarking results, the compiler, and the tools we used to replicate them.

1 Introduction to benchmarks

Benchmarking is a vital tool in the world of semiconductors, enabling developers to measure the performance of a system, software, or hardware. This is particularly important for microcontrollers, which are used in a wide range of applications, from automotive devices to aerospace and defense systems. Two of the most widely used benchmarks for microcontrollers are Dhrystone and CoreMark. These benchmarks have been used for decades and are still in use today, providing a standard way to compare the performance of different microcontrollers.

Dhrystone is a synthetic benchmark that was developed in the early 1980s by Reinhold P. Weicker [1]. At the time, the microprocessor industry was evolving rapidly, and there were several competing microprocessor manufacturers, each claiming their product was superior. Benchmarking provided an objective way to compare the performance of different microprocessors and assess the significant advancements in microprocessor technology. This was done by identifying which new architectures, instruction sets, manufacturing processes, and other technological advancements in microprocessors translated into performance and efficiency improvements. Weicker created Dhrystone as a response to the need for an objective benchmark. It was designed to be a simple benchmark that could easily be implemented and executed on a wide range of microprocessors, and was developed in the C programming language, which was widely supported across different platforms and compilers. Its simplicity and portability made it understandable and accessible to both developers and users, increased its popularity, and made it one of the most widely used benchmarks for microprocessors.

CoreMark is a benchmark that was developed in the early 2000s. It was created in response to the changing landscape of microcontrollers, which had become more complex and capable since Dhrystone was first introduced. It was designed to support multithreaded processors, which had become more prevalent by the time it was introduced. It included specific implementations and test scenarios to evaluate the performance of processors with multiple cores or threads. This allowed CoreMark to assess the efficiency of parallel processing and provide insights into the scalability of a processor's performance. Furthermore, it incorporated a broader range of operations and memory access patterns. It included integer arithmetic, bit manipulation, control flow, and memory access operations, simulating a mix of tasks commonly found in embedded systems and real-time applications. This made CoreMark more relevant for evaluating processors in these specific domains.

The benchmarks are executed on the microcontroller being tested and the results measure the time it takes for the microcontroller to execute the instructions. Dhrystone measures performance by executing a series of string operations, integer arithmetic operations, such as addition, subtraction, bit shifting, and logical operations. CoreMark, on the other hand, is designed to measure performance using a variety of workloads, including matrix manipulation, linked list manipulation, and state machine emulation.

1.1 Benchmark pitfalls

As we are going to see in the following chapters of this white paper, both benchmarks have many fundamental pitfalls which can affect their direct applicability to real-world applications. While these benchmarks provide valuable insights into certain aspects of microcontroller performance, it's important to consider their limitations when drawing conclusions or making decisions based solely on benchmark results.

- Compiler optimizations, inlining decisions, and other compiler settings can lead to variations in benchmark results
- A focus on specific aspects of microcontroller performance such as integer arithmetic, control flow, and memory access are important, but do not capture the full range of factors that impact real application performance
- Limited code size, has no impact on memory hierarchy outside cache

It's important to complement benchmark results with real application profiling, domain-specific benchmarks, and other performance evaluation techniques to gain a comprehensive understanding of a microcontroller's performance in real practical usage scenarios. Therefore, our recommendation is to use the figures below only as an initial indicator of TriCore™ performance.

2 TriCore™ benchmark

TriCore™ is Infineon's core architecture launched in 1999. It was developed to address the perennial quest to achieve more performance in the demanding embedded real-time world. In particular, it found traction in the automotive area, in diverse applications such as engine control, chassis, autonomous driving, etc. Since then, it has been the driver of success stories from the original automotive-focused AUDO to the AURIX™ microcontroller families. Infineon TriCore™ products have been continuously evolving to meet the needs of a growing number of automotive applications.

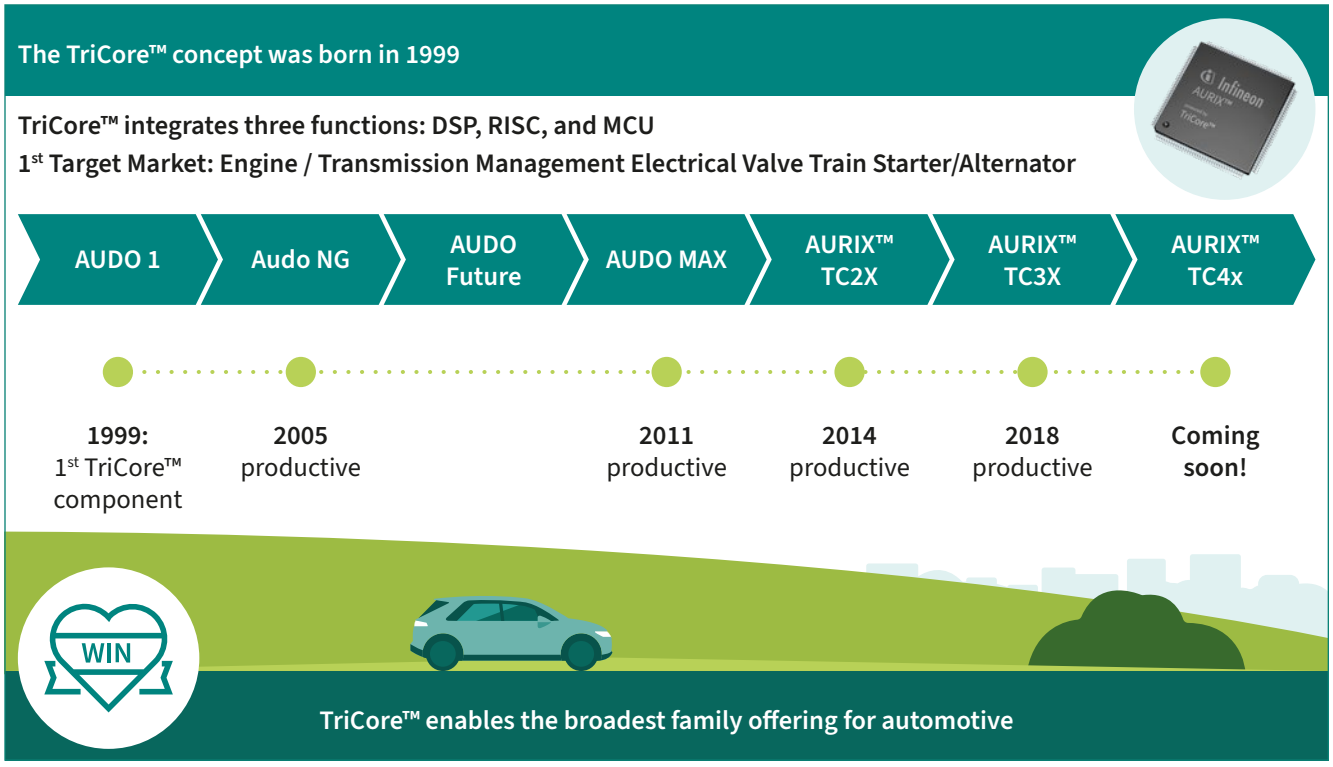

Figure 1 TriCore™ architecture roadmap

2.1 Dhrystone and CoreMark benchmarking

In order to evaluate the performance of TriCore™, we conducted the two widely used benchmark tests, Dhrystone and CoreMark. For the Dhrystone test, we used an optimized GCC 9.4.0 compiler and applied the three different ground rules laid out in the Dhrystone documentation, referred to below as “Ground Zero”, “Ground Inline”, and “Whole Program”:

– The source code was compiled separately, without any use of inline functions (Ground Zero)
– The source code was compiled separately and inline functions were used (Ground Inline)
– The source code was compiled together (Whole Program)

We should also note that the version v2.1 of Dhrystone was used to perform the benchmarking.

Table 1 TriCore™ Dhrystone and CoreMark results

Core	DMIPS/MHz Ground Zero	DMIPS/MHz Ground Inline	DMIPS/MHz Whole Program	CoreMark/MHz
AURIX™ TC3xx TriCore™ 1.6 CPU	2.24	2.74	9.03	5.23
AURIX™ TC4xx TriCore™ 1.8 CPU	2.27	2.76	9.03	5.07

As we see above in Table 1, the Dhrystone performance is impacted and greatly increased by setting the compiler to inline the code, because the compiler attempts to replace function calls with the actual code of the function at the call site, which leads to improved performance by reducing the overhead of function calls. In addition to inlining the code, compiling the Dhrystone program as a whole provides the compiler with more opportunities for optimization, including aggressive inlining, cross-module optimization, constant propagation, dead code elimination, and improved resource usage. In addition to the Dhrystone benchmark, we also performed a CoreMark test and the results are presented in the table above.

Note that the result of the Dhrystone and CoreMark benchmarks is shown as DMIPS/MHz and CoreMark/MHz respectively, and is referred to one TriCore™. So, taking as an example the TC39x device, with six TriCore™ 1.6 cores and a clock frequency of 300 MHz, the total Dhrystone score and CoreMark score is calculated as below:

Dhrystone score (Ground Zero) = 2.24 x 300 x 6 = 4032 DMIPS
CoreMark score = 5.23 x 300 x 6 = 9414 CoreMarks

2.2 Reproducing the results

The GCC 9.4.0 compiler used is publicly accessible on GitHub and the results can be conveniently replicated by using the Infineon evaluation board KIT_A2G_TC397_5V_TF for TC3x, the TC4x TriBoard, and the AURIX™ Development Studio 1.9.4, which offers an easy way to integrate external compilers and is available free of charge here.

3 Conclusion and recommendations

Overall, and as extensively found in public literature [1][2], both Dhrystone and CoreMark have many fundamental pitfalls which reduce their usefulness as good benchmarks to correlate real with test application performance on an MCU, especially for automotive applications. Some of these pitfalls are:

– a very limited code size, which has no impact on memory hierarchy outside cache
– the dependence of much of the code in Dhrystone on string libraries, which is irrelevant to the needs of an automotive application
– and, last but not least, the very high impact of compiler optimizations

Especially the last point is quite important. Hence, while these benchmarks seem to provide useful information about microcontroller performance, they are actually only weakly correlated to real application performance. Therefore, our recommendation is to use the above numbers only as an initial indicator of TriCore™ performance while relying on application-based benchmarks to get a better picture of TriCore™ performance.

References

[1] R. York, ARM Ltd, “Benchmarking in context: Dhrystone”, 2002
[2] A. R. Weiss, “Dhrystone Benchmark: History, Analysis, “Scores” and Recommendations”, 2002

Published by
Infineon Technologies AG
Am Campeon 1-15, 85579 Neubiberg
Germany

Public

Date: 01/2024

Stay connected!

Scan QR code and explore offering
www.infineon.com

Please note!
This Document is for information purposes only and any information given herein shall in no event be regarded as a warranty, guarantee or description of any functionality, conditions and/or quality of our products or any suitability for a particular purpose. With regard to the technical specifications of our products, we kindly ask you to refer to the relevant product data sheets provided by us. Our customers and their technical departments are required to evaluate the suitability of our products for the intended application.

We reserve the right to change this document and/or the information given herein at any time.

Additional information
For further information on technologies, our products, the application of our products, delivery terms and conditions and/or prices, please contact your nearest Infineon Technologies office (www.infineon.com).

Warnings
Due to technical requirements, our products may contain dangerous substances. For information on the types in question, please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by us in a written document signed by authorized representatives of Infineon Technologies, our products may not be used in any life-endangering applications, including but not limited to medical, nuclear, military, life-critical or any other applications where a failure of the product or any consequences of the use thereof can result in personal injury.