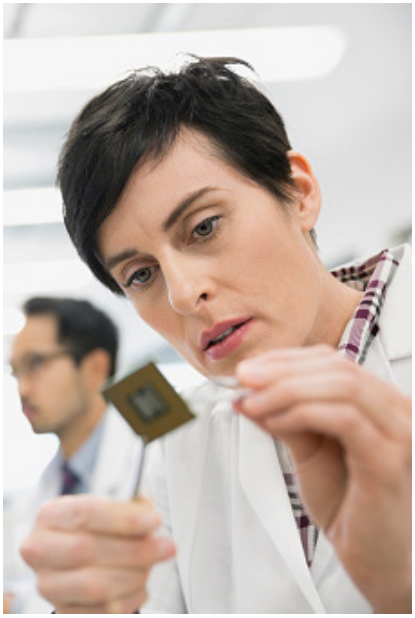
An Introduction to CPU Performance Benchmarks and How This Applies to the Home Market

arm

November, 2021

Whitepaper



Abstract

Benchmarks are a vital tool for measuring the performance of CPUs in the most popular consumer devices, particularly in the Home segment which includes digital televisions (DTVs) and set-top boxes (STBs).
However, benchmarks are evolving at a rapid pace, moving from standalone measurements to considering real-world use cases to gain a more accurate representation of performance. This whitepaper provides details about recent developments in performance benchmarks on CPUs and how it has evolved from the early benchmarks of the 80s. It also outlines the benefits and challenges of using the various available benchmarks to measure CPU performance on Home and other consumer devices.

An Introduction to Benchmarks

Many different metrics have been used in the past 35 plus years to measure performance of a processor – MIPS, MOPS, MFLOPS, and MHz. Dhrystone, developed in 1984, was an early attempt to measure performance by executing real code. It served the industry for a long time, but, due to its synthetic nature, toy-sized footprint, and susceptibility of performance hot-spots to toolchain gaming, the benchmark became irrelevant as a measure of prevailing real-world application performance.

CoreMark was then created by the Embedded Microprocessor Benchmark Consortium (EEMBC) as a simple and standardized embedded benchmark that addressed some of the limitations of Dhrystone, while retaining its ease of use in porting and running on small devices.

Since the early days of Dhrystone and CoreMark, other benchmarks have been developed to measure performance of application class processors. These benchmarks have been updated periodically to keep them up to date with evolving application characteristics and growing complexity. They cover a wider range of applications, including programming languages, compilers, combinatorial optimization, audio and video compression, and Artificial Intelligence (AI), that more accurately represent the types of workloads that run on modern classes of processor.

Today, real world applications running on larger application class processors exhibit a great diversity of characteristics in their code and data footprints, control flow, memory access patterns, use of multi-threading and multi-processing, and use of system peripherals. Therefore, a single benchmark program is unable to represent the entire spectrum of diverse real-world applications and use cases. However, more modern and complex benchmarks can stress the processor core pipeline and memory system in diverse ways compared to earlier benchmarks like Dhrystone and Coremark.

When selecting benchmarks, the challenge for the industry is the wide spectrum of performance and overlap between embedded and application processor performance points that can make it hard to work out the fairest way to measure processor performance. This paper discusses the underlying requirements for a benchmark and why trying to use the wrong measure will produce misleading results.

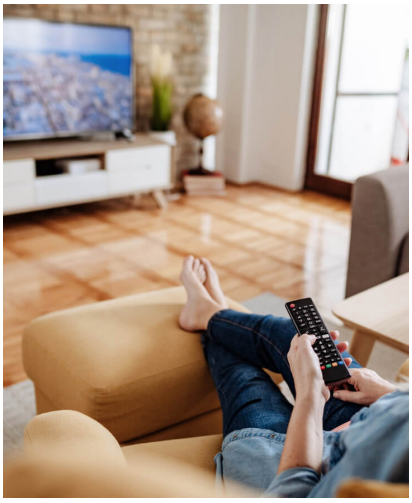
Benchmark Characteristics

Processor architects make different trade-offs between several design factors, such as the clock frequency, core pipeline, degree of out-of-order execution, number of execution units, cache organization and size, memory hierarchy, etc. The performance of a processor is determined by the microarchitecture bottlenecks (design trade-offs) and how they are exercised by the application program (application characteristics).

Benchmark suites are a proxy, so one way of assessing their representativeness is to compare their performance characteristics to a range of real-world application programs. For single-threaded programs, the characteristics and bottlenecks that determine the manifested performance are the stress on the core pipeline, instruction cache, data cache, memory, and branch prediction. These characteristics can be measured, normalized, and visualized per application. The visualizations are in some sense a performance signature of the application or benchmark. This performance signature, along with the inherent knowledge of the application or benchmark, can be used to compare the similarities and differences across them.

At Arm, we measure key performance attributes that quantify the stress on the core pipeline, instruction cache, data cache, memory, and branch prediction for a range of benchmarks, tasks and applications on our Cortex CPU processors. Meanwhile, the range of applications include -

boot up sequence; application launch; compression; AI; compilers and interpreters; path-finding algorithms; combinatorial optimization; discrete event simulation; and parsers.



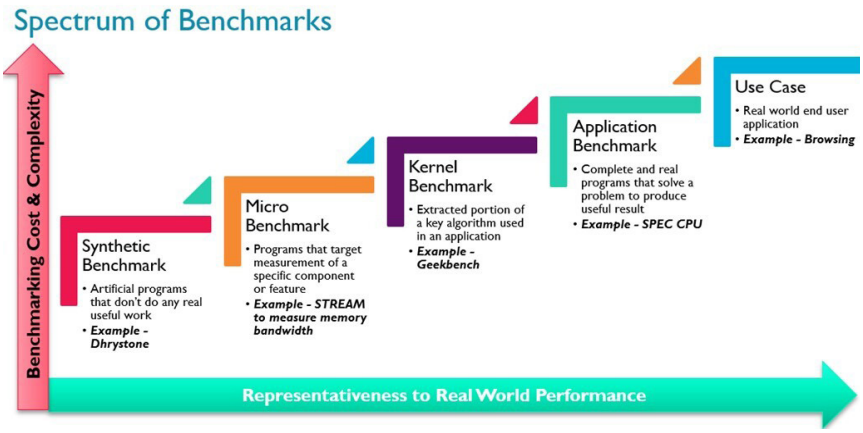
Arm CPUs in the home segment market

Within each segment of consumer devices, different benchmarks are needed to a) reflect the different use cases and workloads and b) reflect the different CPUs that are used. The segment market for home consumer devices is an interesting case study for benchmarks. Arm technology is currently deployed in over 600 million new home consumer devices every year, from Smart TVs to STBs, game consoles and smart displays. The fast growth of content providers and streaming services has driven the demand for improved technologies in this segment. Arm offers a complete solution in this space, ranging from CPUs and GPUs to System IP and dedicated neural network units.

Devices built on Cortex-A CPUs and Mali GPUs make up a sizeable number of the home segment market. Cortex-A CPUs deliver smooth and vivid visuals for the best screen experience. Mali GPUs deliver superior graphics, enabling key use cases like gaming and 4K UI. Moreover, Arm's System IP enables an easier integration improving energy efficiency and system performance.

In the home segment, a variety of CPUs that vary in terms of performance and efficiency are used due to the nature of the market and the continuous drive towards more complex use cases (e.g., super resolution, AI, etc.). Therefore, now more than ever, it is important to understand the spectrum of different benchmarks and how they measure overall system performance.

The spectrum of benchmarks



The figure above shows the entire spectrum of benchmarks used within Arm. The key categories are synthetic, micro, kernel, application, and use case-based benchmarks. As we move up and to the right on the chart, the benchmarks start having a higher level of correlation with the end user experience and more of a system play. However, the cost and complexity to set them up, maintain, and use for pre-silicon analysis also starts significantly increasing.

At one end of the spectrum are synthetic benchmarks, such as Dhrystone and CoreMark. They are not very representative of real-world performance, but are quite easy to port, run, maintain, and do not require a full system setup for projecting scores.

Next on the spectrum are microbenchmarks that are typically used to measure specific aspects of IP performance, such as memory latency and bandwidth. For example, the STREAM benchmark is used specifically to measure sustained memory bandwidth.

Kernel benchmarks comprise of extracted algorithms from a range of algorithms and can help assess performance on key hotspots from larger applications. Geekbench is one example of a kernel benchmark, broadly used in mobile and desktop applications. The latest version is Geekbench 5, which is 64-bit only. It uses a scoring system that separates single core and multi core performance.

Application benchmarks comprise of complete and real programs that are widely used to solve various compute challenges but could be complex to port and setup. One example is SPEC CPU, which is the most popular benchmark for measuring CPU performance. The latest version has been updated to be more representative of real-world applications.

Finally, at the end of the spectrum there are use case-based benchmarks. Speedometer is one example of a use case-based benchmark that measures the responsiveness of web applications. These are very representative of end usage, but difficult to port, run, maintain, and require full-system platforms for projecting scores. Also, the metrics to measure the performance of use cases – JANK, frame rate, frame drops, etc. – are difficult to measure in pre-silicon platforms. This makes it difficult to use them for performance exploration in the early stages of the design cycle. Moreover, use case-based benchmarks can often be impractical as a measure of IP performance. For example, a web browsing benchmark will exercise a complex software stack, system, and CPU IP. Although especially useful for analysing the performance of a product, it is difficult to isolate the performance upside from improvements in the software, system, and CPU.

Each of the categories of benchmarks has its own place and value during various stages of the hardware and software product design stage. For example, synthetic benchmarks such as Dhrystone can be used as a proxy to represent the power of longer running benchmarks; microbenchmarks, such as STREAM, are used to measure peak achievable bandwidth; kernel benchmarks provide an estimate of performance of specific algorithms; and application/use case-based benchmarks run late in the design cycle and provide an accurate measure of system level performance.

It is important to note that the list of benchmarks continues to cumulatively evolve over time and needs to be kept relevant to the segments targeted by Arm products. The current portfolio of benchmarks is not necessarily exhaustive in terms of: (a) representativeness to performance characteristics of emerging workloads/segments (e.g., high instruction cache miss-rate is observed in server workloads but not in the existing suite of benchmarks); (b) assessing benefits of new instructions incorporated in the architecture (e.g., assessing performance benefit of LDAPR, Load Acquire-RcPC register, instruction); and (c) evaluating design alternatives and assessing goodness of microarchitecture features for emerging workloads (e.g., instruction-

- cache side improvements for server workloads) and newer architecture features (e.g., SVE).

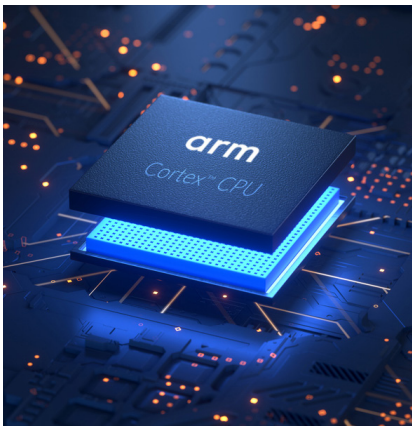
It is also important to note that the choice of benchmarks used for pre-silicon projections and post-silicon benchmarking, particularly for CPUs, is driven by factors that include partner requests, popularity in the external ecosystem, competitive pressures, and technical merit of benchmark.

Benchmark	Organization	Description	Comment
Dhrystone	Open source	Synthetic benchmark measure compute-intensive integer performance.	No longer representative of modern workloads but continues to be popular for legacy reasons
CoreMark	EEMBC	Popular benchmark to measure compute-intensive integer performance. This benchmark addresses lacunae of Dhrystone (compiler gaming and performance sensitivity to system libraries).	Results can be published or certified through EEMBC.
SPEC CPU	SPEC OSG CPU	Application benchmarks that comprise of open-source real world applications. Sixth generation of SPEC CPU benchmark suite – SPEC CPU2017 is the latest version- it was released in 2017	Most popular benchmark for measuring CPU performance. Latest iteration of benchmark representative of prevailing real-world applications.
LMBench 3.0	Open Source	Suite of microbenchmarks to measure various aspects of memory latency and bandwidth.	Measure of memory latency and bandwidth.
STREAM	Open source	Microbenchmarks to measure sustained memory bandwidth.	Measure of sustainable memory bandwidth.
Geekbench 5	Primate labs	Kernel benchmarks that measure CPU Integer, Floating Point, and Memory performance.	Used in industry, but with some limitations on the representation of real-world scenarios.

Conclusions

To be useful, a benchmark must be representative of the characteristics of the target real world applications and use cases. In the past 35 years, benchmarks have evolved from standalone performance measurements to being focused on real-world use cases. Using an application benchmark, such as SPEC CPU, instead of Dhrystone and Coremark is a huge step forward in benchmarking an application class processor. Therefore, our assertion is that if a workload needs an application class processor, then older benchmarks such as Dhrystone and Coremark are too simple and, as a result, not appropriate to usefully measure this. This is important because trying to compare devices using an unsuitable benchmark may lead to disappointment when running your real application code. For example, techniques to improve compute-bound synthetic benchmarks, such as Dhrystone, will -

not necessarily translate into improvements in real-world applications that are likely to be more memory bound.



However, it should be noted that any benchmark suite will only represent a portion of the performance spectrum of real-world usages. The industry should rely on a range of benchmark suites to measure a wide range of applications and performance points. In fact, there can still a role for older benchmarks that purely measure the compute-bound performance of synthetic programs, but these are best deployed in the early stages of processor and IP development. Using these older benchmarks combined with application and use case-based benchmarks can provide useful performance insights during the early development stages of IP.

An approach that would complement the existing portfolio of benchmarks would be to develop a methodology to distil the essence of use cases into microbenchmarks or kernel benchmarks with similar stress patterns as the use cases. These microbenchmarks can be used early in the design cycle for stressing key aspects in existing IPs, project high-level metrics early in the design cycle, and create a suite for performance verification.

Finally, it is worth noting that different benchmark suites might be more applicable to certain device types and market segments. For example, processors and IP for consumer devices in the home segment will greatly benefit from application benchmarks like SPEC CPU that will be able to more effectively measure the range of more complex workloads that are now happening on these devices. However, some older benchmarks might still be applicable for consumer devices that do not run complex applications or have a very high compute processing requirement. Benchmarks are not a 'one-size-fits-all' for all processors, devices, and use cases, so it is worth reflecting on the information in this whitepaper when selecting the most appropriate benchmark to use.