An Independent Evaluation

of the

Cadence Tensilica Fusion G3 DSP Core

By the staff of

BDTi

October 2016

OVERVIEW

The recently announced Cadence Tensilica Fusion G3 DSP IP core is a high-performance licensable programmable digital signal processor core targeting diverse signal processing applications such as communications, audio and industrial applications.

BDTI, a technology analysis firm, benchmarked the Fusion G3 core on several typical digital signal processing functions, comparing the performance of Fusion G3 against Texas Instruments' flagship C66x DSP core. BDTI also compared the Fusion G3's FFT performance to that of the ARM Cortex-A57 CPU core. Finally, BDTI implemented and optimized a custom DSP function from scratch on the Fusion G3 and compared the resulting performance to that of the TI C66x. This report presents BDTI's independent evaluation of the Fusion G3 core's performance and ease of software development.

The Fusion G3 DSP core's wide SIMD (single-instruction, multiple-data) operations and VLIW (very long instruction word) instruction set provide excellent cycle efficiency on many DSP tasks, and yield performance that surpasses that of TI's flagship C66x DSP core. Fusion G3 is also noteworthy for its double-precision floating-point support for precision-critical tasks. Cadence provides robust software development tools and DSP function libraries to help users effectively realize the core's performance potential.

© 2016 Berkeley Design Technology, Inc.

Contents

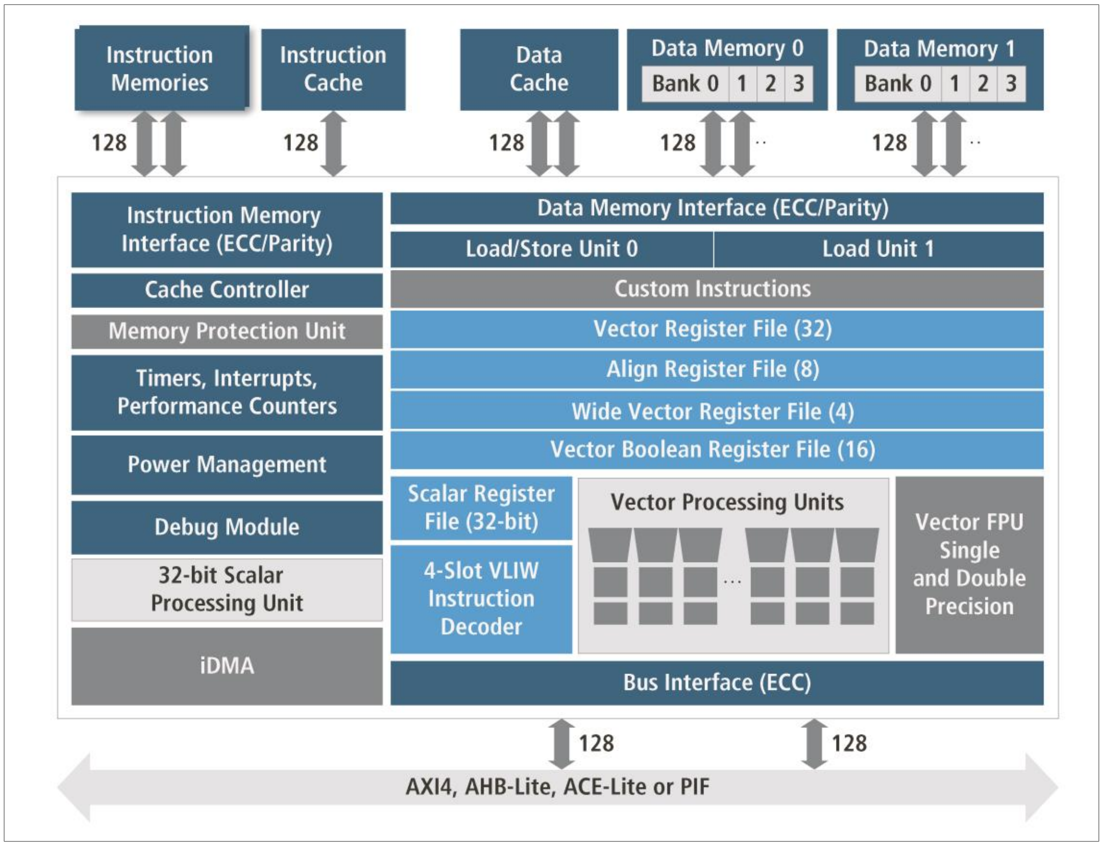
1. Introduction

The Cadence Tensilica Fusion G3 DSP is a high-performance programmable digital signal processor core from Cadence, targeting a wide range of signal processing workloads in applications such as communications, audio, and industrial equipment. This report presents BDTI's independent evaluation of the Cadence Fusion G3 DSP core's performance and ease of software development.

We compare Fusion G3's execution speed and cycle-efficiency against the Texas Instruments' flagship C66x high-performance DSP core, used in many of TI's DSP chips. BDTI chose the TI C66x for this competitive evaluation because it targets a similar range of applications, is well known in the industry, and has readily-available tools and optimized libraries that we could leverage for benchmarking. We used existing, optimized software library functions available from Cadence and TI, respectively, to compare the performance of the Fusion G3 and C66x.

We also compare the Fusion G3's execution speed on floating-point FFTs against ARM's Cortex-A57 core leveraging ARM NEON SIMD instructions. The Cortex-A57 is a licensable 64-bit high-performance CPU core. Comparing Fusion G3 performance against that of the Cortex-A57 illustrates the benefits of including a dedicated DSP core in a SoC design. However, CPU cores such as Cortex-A57 are typically used with different system-level hardware architectures, different software architectures, and different programming practices compared with DSP cores, making performance comparisons between these two classes of processors challenging and prone to misinterpretation. Therefore, we consider the comparison of Fusion G3 and Cortex-A57 to be a rough first-order comparison.

Finally, we discuss BDTI's software development and optimization experience with



Instruction Memories
Instruction Cache
Data Cache
Data Memory 0
Bank 0 1 2 3
Data Memory 1
Bank 0 1 2 3
128
128
128
128
128
Instruction Memory Interface (ECC/Parity)
Cache Controller
Memory Protection Unit
Timers, Interrupts, Performance Counters
Power Management
Debug Module
32-bit Scalar Processing Unit
iDMA
Data Memory Interface (ECC/Parity)
Load/Store Unit 0
Load Unit 1
Custom Instructions
Vector Register File (32)
Align Register File (8)
Wide Vector Register File (4)
Vector Boolean Register File (16)
Scalar Register File (32-bit)
4-Slot VLIW Instruction Decoder
Vector Processing Units
Vector FPU Single and Double Precision
Bus Interface (ECC)
128
128
AXI4, AHB-Lite, ACE-Lite or PIF

Figure 1 Block Diagram of Fusion G3 Architecture

the Fusion G3 core and toolchain. In addition to using DSP functions from Cadence's DSP library, BDTI implemented and optimized on the Fusion G3 a median filter common in video processing applications. In Section 5 we describe and comment on our experience with function optimization on Fusion G3 and with Cadence software libraries and tools.

2. About the Cadence Fusion G3 Core

Cadence positions the Fusion G3 DSP core as a multi-purpose DSP for compute-intensive applications. The Fusion G3 is suitable for diverse workloads including communications, audio, imaging, radar and industrial control, among others. A block diagram of the Fusion G3 architecture is shown in Figure 1.

Fusion G3 natively supports 8-bit, 16-bit, and 32-bit fixed-point arithmetic via 128-bit wide SIMD (single-instruction, multiple-data) vector registers, performing parallel operations on sixteen, eight, or four data elements at a time, respectively. An optional vector floating-point unit adds support for SIMD operations on 32-bit single-precision and 64-bit double-precision floating-point data, compliant with IEEE 754 standards.

A VLIW (very long instruction word) instruction set architecture allows the Fusion G3 core to execute up to four SIMD vector operations per cycle, including (with some restrictions) up to two multiplier/ALU operations and two loads or one load and one store. Flexible operation predication support allows vectorized code to maintain high throughput by avoiding conditional branches.

3. About the Benchmarks

Common DSP Functions

BDTI measured the performance of the Cadence Tensilica Fusion G3 DSP core and the Texas Instruments C66x DSP core on several functions from the optimized software libraries available from Cadence and Texas Instruments. We believe that these software libraries are widely used, and their performance generally reflects real-world application performance.

BDTI benchmarked the DSP algorithm kernel functions listed in Table 1. For each function, we measured the number of processor cycles taken by the Fusion G3 and C66x to execute the function. We computed execution time for each target by dividing the measured number of cycles by the processor's clock rate. For the Cadence Fusion G3 we measured cycle counts using a cycle-accurate simulator, and for the C66x we used an evaluation board.

The benchmark functions were chosen based on relevance to a variety of signal processing tasks, and also based on availability in a comparable form on both the Fusion G3 and C66x platforms. The Cadence and Texas Instruments optimized DSP function libraries both contain multiple implementations of some algorithm kernel functions such as FIR filters. Different implementations impose different restrictions on function parameters, for example requiring that the number of filter taps be a multiple of four. Such restrictions represent tradeoffs between function generality and optimal vectorized implementation. BDTI made best efforts to choose the fastest available function for each target processor, while also ensuring that the restrictions imposed by the respective implementations are reasonably comparable. Thus, the requirement of availability of comparable implementations on both targets restricted the choice of algorithm kernels somewhat.

Benchmark	Parameters	Data Types
Complex FFT	512 points	16-bit fixed-point, 32-bit floating-point
Real FIR Filter	32 taps, 1024 points	16-bit fixed-point, 32-bit floating-point
Real Vector Dot Product	1024 points	16-bit fixed-point, 32-bit floating-point
2D Median Filter	3×3 kernel, output one scan line, 1280 pixels	8-bit unsigned
Matrix LU Decomposition	32×32 matrix	64-bit floating-point

Table 1 DSP functions selected for benchmarking the Fusion G3 and C66x. For each function, BDTI benchmarked an optimized implementation for each data type listed.

The LU matrix decomposition algorithm kernel was chosen specifically to highlight the support for double-precision floating-point arithmetic on the Fusion G3 core. Matrix decomposition and inversion functions often require double-precision floating-point arithmetic. Of the algorithm kernels provided in the Cadence and TI libraries that commonly require double-precision floating-point support, the LU decomposition kernel functions have the most closely comparable APIs between the Fusion G3 and C66x DSP libraries. Note, however, that while the TI matrix LU decomposition function generates the permutation matrix as one of its outputs, the corresponding Cadence function does not. BDTI estimated the additional number of cycles that would be required by the Cadence function to output the permutation matrix. The number of cycles for the Fusion G3 on the matrix LU decomposition reported here is computed by adding the measured cycles and this small estimated overhead.

All functions were benchmarked on both processors with all input data, coefficients, twiddle factors, and output data in tightly-coupled fast SRAM, minimizing load/store latencies. This is necessary in order to prevent differences in memory system architecture from skewing the benchmark results. Therefore, the results reflect use cases in which each benchmarked function is time-critical to the application, and the programmer has taken special care to utilize fast local memories.

To compute execution time, BDTI used a Cadence-provided clock rate estimate of 1.1 GHz for the Fusion G3. BDTI believes that the 1.1 GHz clock rate estimated by Cadence is readily achievable by licensees, either by choosing a more advanced mainstream fabrication process, or by optimizing their SoCs for performance. For the C66x we used a clock rate of 1.25 GHz, the highest rate at which the DSP can be clocked in most C66x devices. While the C66x is available in chip form at several speed grades including 1.25 GHz, the Fusion G3 is a licensable core and its clock rate in a physical device will depend on many factors including process technology, cell library, floor planning, speed binning, and so on. Therefore, the performance comparison presented here should be viewed as representative but not guaranteed.

Evaluating Ease of Software Development

The median filter function was chosen for the purpose of evaluating software development user experience for the Fusion G3. This function was attractive because it is not included in the Cadence function library—thus requiring custom implementation and optimization—and because there is a median filter implementation available in the Texas Instruments image processing software library for the C66x core, enabling performance comparison against the C66x. BDTI created and optimized a median filter implementation for Fusion G3 that is designed to be meaningfully comparable to the median filter function found in the Texas Instruments library. Therefore, the median filter benchmark result presented in this document reflects a fair comparison of the Fusion G3 and C66x core architectures, rather than the fastest possible implementation of a median filter.

Comparing Fusion G3 to ARM Cortex-A57

To compare the performance of Fusion G3 against that of an ARM CPU core, BDTI measured the execution time of single-precision floating-point complex FFT on ARM Cortex-A57. Many fundamental differences between embedded DSP cores such as the Fusion G3 and CPU cores such as the Cortex-A57 conspire to make performance comparisons challenging. Because of the dramatic differences in both the core architectures and their typical use, the results presented here can only be considered a rough first-order comparison between the Fusion G3 and Cortex-A57. The most noteworthy challenges in comparing Fusion G3 to Cortex-A57 are described in the Appendix.

4. Benchmark Results

Overall, benchmark results indicate that a Fusion G3 core running at 1.1 GHz performs substantially better than Texas Instruments' flagship C66x DSP running at 1.25 GHz. A 1.1 GHz Fusion G3 core also handily outperforms an ARM Cortex-A57 CPU running at 1.7 GHz. Additional detail and analysis are presented below.

Fusion G3 vs. C66x

Figure 2 shows the execution time of Fusion G3 relative to the Texas Instruments C66x DSP core, for each of the benchmark functions. The Fusion G3's relative performance ranges from 8%

faster on the 16-bit fixed-point FFT, to roughly 5.3 times faster on the double-precision matrix LU decomposition, compared to the C66x. On average, the Fusion G3 is 34% faster than the C66x on this set of benchmarks. However, considering that double-precision floating-point arithmetic support is less frequently required than the other data types represented in the benchmarked functions, it may be desirable to treat the matrix LU decomposition benchmark as an special case. Averaging the remaining benchmarks, the Fusion G3 is 27% faster than the C66x.

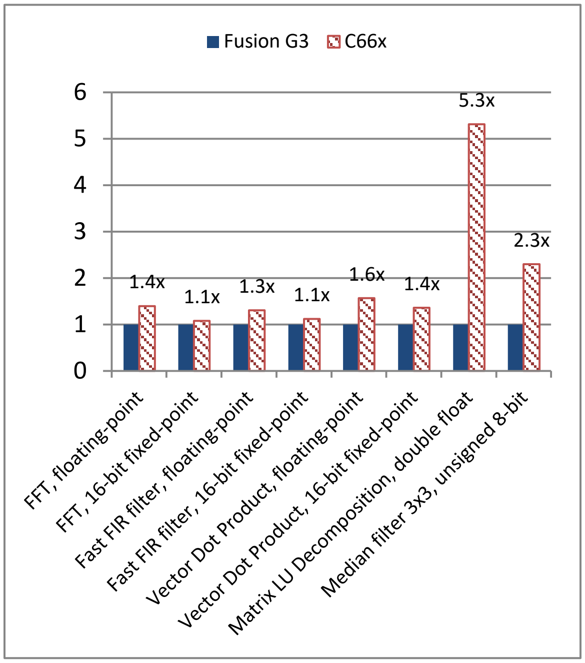


Figure 2 Fusion G3 vs C66x Execution Time (lower is better)

Combining the three single-precision floating-point benchmark results (FFT, FIR, and vector dot product), the Fusion G3 execution time is 29% faster on average than the C66x on floating-point tasks. Combining the three 16-bit fixed-point results (FFT, FIR, and vector dot product), the Fusion G3 execution time is 15% faster on average than the C66x on 16-bit fixed-point tasks.

For applications that require double-precision floating-point support, the matrix LU decomposition result suggests that the Fusion G3 provides an overwhelming advantage. When double-precision arithmetic is critical to the application, further benchmarking is recommended in order to validate this advantage for a wider range of functions.

The Fusion G3's substantially greater cycle efficiency more than compensates for its lower operating frequency to yield execution speed that clearly outperforms the C66x. The Fusion G3's benchmark cycle counts relative to C66x are shown in Figure 3. Relative to the C66x, the Fusion G3 core ranges from 18% more cycle-efficient on the 16-bit fixed-point FFT to six times more cycle-efficient on the double-precision matrix LU decomposition.

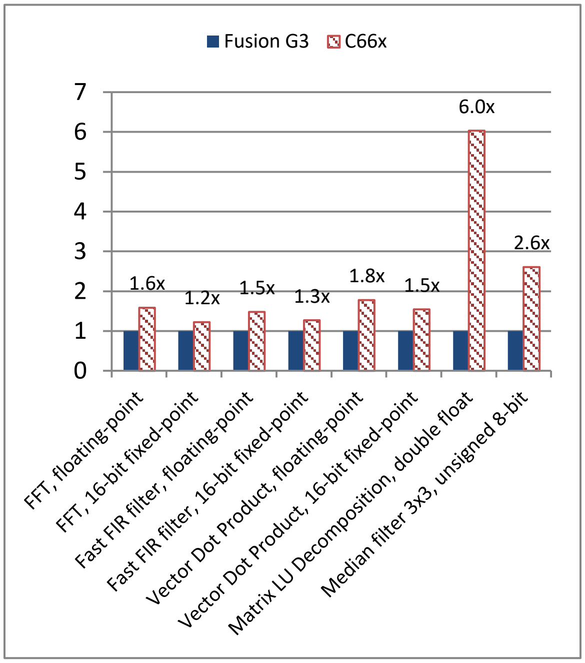


Figure 3 Fusion G3 Benchmark Cycles Relative to C66x (lower is better)

Fusion G3 vs. ARM Cortex-A57

As discussed in Section 3 above, we compared the execution speed of Fusion G3 to that of ARM Cortex-A57 only on the floating-point FFT function. As shown in Figure 4 , the Fusion G3 at 1.1 GHz completes the 512-point complex FFT in about 2.02 µs, more than two times faster than the 4.8 µs measured on the ARM CPU at 1.7 GHz. Although we believe that additional optimization might yield a faster FFT on the ARM Cortex-A57, we expect that, to a first order, the result presented here is representative of typical use as described in Section 3. These FFT execution speed results demonstrate a clear advantage of Fusion G3 over an ARM CPU, and we expect to see similar advantages on other signal processing functions.

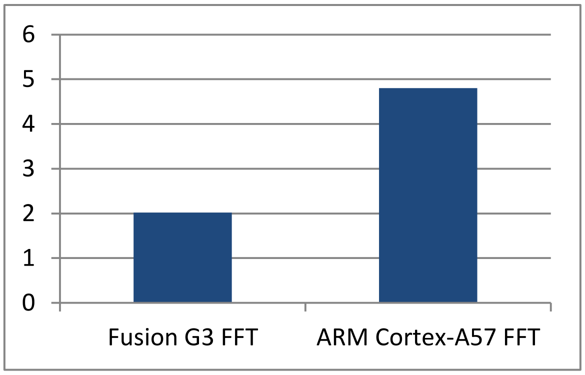


Figure 4 Fusion G3 vs. ARM Cortex-A57 FFT Execution Time in µs (lower is better)

5. BDTI's Fusion G3 Software Development User Experience

In this benchmarking effort, BDTI used the software development tools for the Fusion G3 core to create simple test bench executables that validate the functionality and measure the cycle counts of functions in the Cadence-provided DSP library. BDTI also implemented, optimized, and benchmarked a custom median filter function as described in Section 3 above. We used only the Windows version of the tools, although a Linux version is also available. The Fusion G3 tools include the Xtensa Xplorer IDE—a common IDE used with all Cadence Tensilica Xtensa cores including Fusion G3. The IDE integrates GNU-based build and debug tools and a cycle-accurate simulator. A multiprocessor modeling environment is also included, but we did not use it in this benchmarking effort.

Overall we found the Cadence toolchain to be complete, robust, and easy to use. After installation, adding a pre-release version of the Fusion G3 core configuration required some guesswork and ultimately a bit of support from Cadence, but we expect that this will be fixed in an official release. Once installed and properly configured, the tools have a familiar feel for developers accustomed to the IDEs provided by other DSP processor vendors. File, workspace, and project hierarchies are easy to understand and use. The tools are stable and libraries function and perform as documented. We particularly liked the import/export functionality of the Xplorer IDE, which we've found helpful for sharing source code among engineers, along with corresponding compiler and linker options. One inconvenience arose when trying to save the results of a profiler run via the IDE: Our first attempts to specify the desired profiler command line arguments via the IDE did not work, and we switched to executing the profiler from the DOS command line to bypass the issue.

The DSP function library for the Fusion G3 is well documented and thoroughly optimized. We encountered no issues in our benchmarking of the library functions we selected for this work. In contrast, the DSP libraries from TI required some guesswork, source code inspection, and experimentation in order to get correct functionality and peak performance. Compared to the Cadence DSP library, the TI library was also more prone to saturation and overflow, and required scaling some of the data, resulting in lower precision output.

The matrix LU decomposition function in the Cadence DSP library closely matched the decomposition generated by MATLAB. In contrast, the respective function in the TI DSP library found a different decomposition of the input matrix that was valid but did not match MATLAB's result.

For a deeper evaluation of the software development experience on Fusion G3, BDTI implemented and optimized a median filter function. To take advantage of the Fusion G3's computational horsepower, developers must optimize the use of SIMD vector registers and operations, and ensure that the four slots in the core's VLIW pipeline are efficiently utilized. The tools can perform automatic vectorization of C/C++ code. To ensure best performance, BDTI chose to bypass this feature and code the median filter function using C with compiler intrinsics. Although writing hand-coded assembly is theoretically possible, it is generally impractical for complex VLIW processors such as Fusion G3 and TI's C66x. Excellent results are possible using compiler intrinsics, and Cadence discourages writing hand-coded assembly for its VLIW processors.

The Fusion G3 DSP core provides a rich set of instructions, requiring a modest learning curve to get started with intrinsic-based optimization. A few ambiguities in the documentation of instructions required some experimentation and inspection of example code from Cadence in order to correctly utilize the compiler intrinsics. We were also occasionally puzzled by cryptic or counter-intuitive compiler warnings. Cadence plans to fix these issues in future releases of the toolchain. Aside from these minor annoyances we found intrinsic-based optimization to be very effective and we reached good cycle-efficiency with reasonable effort for a high-performance DSP.

6. Conclusions

BDTI benchmarked the Cadence Tensilica Fusion G3 DSP core and compared its execution speed to that of the Texas Instruments C66x DSP core using several common DSP functions and utilizing several different data types. The Fusion G3 is significantly more cycle-efficient than the C66x on every benchmark, resulting in an overall execution speed advantage for the Fusion G3 even at a lower clock rate than that of the C66x. The Fusion G3 core is noteworthy for its particularly strong support for double-precision floating-point arithmetic.

BDTI also compared the execution speed of Fusion G3 to that of the ARM Cortex-A57 CPU, using floating-point complex FFT as a benchmark. This rough first-order comparison suggests that a 1.1 GHz Fusion G3 core will handily outperform a 1.7 GHz ARM Cortex-A57 core, illustrating the advantage of offloading DSP functions from a CPU to a Fusion G3 core.

Finally, BDTI evaluated the ease of software development on the Fusion G3 core by writing and optimizing a median filter function for the Fusion G3. Utilizing the Fusion G3's SIMD and VLIW capabilities via compiler intrinsics proved to be an effective methodology, yielding good performance with reasonable effort. The Cadence-provided IDE is stable, complete, and feels familiar to experienced embedded DSP programmers. The Cadence-provided DSP library for Fusion G3 simplifies software development with optimized implementations of common DSP functions that are robust and well-documented.

Strong DSP performance for both fixed- and floating-point data types combined with high-quality tools make the Cadence Tensilica Fusion G3 DSP core a compelling choice for chips targeting compute-intensive DSP applications.

Appendix: Fusion G3 vs. ARM Cortex-A57 Benchmarking Considerations

The Fusion G3 and Cortex-A57 are very different types of processors, designed for different purposes and with dramatically different architectures. Many fundamental differences between embedded DSP cores such as the Fusion G3 and CPU cores such as the Cortex-A57 conspire to make performance comparisons challenging. These differences include the use of data caches, multi-core clusters, operating systems, and software development practices and tradeoffs. Therefore, although BDTI made reasonable efforts to measure comparable workloads on Fusion G3 and Cortex-A57, we consider the results presented here to be only a rough first-order comparison.

The FFT function was chosen for this purpose because the FFT is a very common DSP application building block, and because FFT implementations are somewhat more amenable to fair comparisons across different architectures. Other benchmark functions that we considered in this project were subject to more-varied programming practices and assumptions when implemented on Fusion G3 vs. Cortex-A57. In order to avoid potentially misleading results, comparisons against ARM Cortex-A57 were limited to floating-point FFT only.

The number of complex FFT points was chosen so that input, output, and twiddle factors fit in the Cortex-A57's 32 kbyte data cache, to make performance measurements as comparable as possible to the Fusion G3 measurements, where fast local SRAM is used for all data and twiddle factors.

BDTI measured the performance of several different FFT functions on ARM Cortex-A57:

- Ne10 library FFT optimized with hand-coded assembly using the ARMv7-A 32-bit instruction set architecture.
- Ne10 library FFT optimized with compiler intrinsics targeting the ARMv8-A 64-bit instruction set architecture.
- Cortex-A57-specific ARM Performance Libraries build, single-threaded library (ARMv8-A 64-bit instruction set).

Theoretically, use of the ARMv8-A 64-bit instruction set architecture reduces register pressure and should enable significantly faster FFT implementation compared to the ARMv7-A 32-bit instruction set architecture. However, the 32-bit ARMv7-A FFT implementation from the Ne10 library yielded the fastest execution time for 512-point complex FFT on Cortex-A57 in our tests, probably due to more aggressive hand-coded assembly optimization. Therefore:

- FFT execution time on ARM Cortex-A57 presented in this document is based on the Ne10 library targeting the ARMv7-A instruction set and running in 32-bit user space.

- We believe that faster execution time is possible using the ARMv8-A instruction set. However, we also believe that in practice developers will typically utilize available libraries rather than hand-optimize their own FFT routines. Therefore we believe that the execution time reported in this document is the fastest that developers are likely to attain in practice.

Also note that ARM Cortex-A57 CPUs are typically found in multi-core devices where they are coupled with Cortex-A53 cores in a big.LITTLE configuration. The Cortex-A53 cores are slower but provide much better power efficiency, allowing the more power-hungry Cortex-A57 cores to be turned off when the system is not heavily loaded. Therefore, comparing execution time of Fusion G3 against Cortex-A57 represents a speed-critical use case rather than a power-optimized use case. For a power-efficiency comparison, it would be more appropriate to compare the Fusion G3 vs. the Cortex-A53, and therefore power-efficiency comparison of Fusion G3 vs. ARM CPUs cannot be meaningfully extrapolated from the results presented here.