ACCEPTED MANUSCRIPT

Title: Comparison of computational efficiency of various implementations of FFT algorithm on DSP processor

Authors: Jakub Banach, Michał Śmieja

To appear in: Technical Sciences

Received 9 February 2026;

Accepted 19 March 2026;

Available online 22 April 2026.

*Corresponding author, e-mail: jakub.banach@uwm.edu.pl

COMPARISON OF COMPUTATIONAL EFFICIENCY OF VARIOUS IMPLEMENTATIONS OF FFT ALGORITHM ON DSP PROCESSOR

Jakub Banach[1], Michał Śmieja[2]
[1]ORCID: 0000-0002-9764-2549
[2]ORCID: 0000-0002-9024-3289
Faculty of Technical Sciences
University of Warmia and Mazury in Olsztyn, Olsztyn, Poland

Abstract

The ongoing development of hardware solutions enabling more efficient signal processing is influencing the development of capabilities for reasoning about the state of technical objects. At the same time, solutions for processing data in close proximity to the technical object – edge computing – are becoming more widespread. In an era of new possibilities and expectations regarding signal processing, it is necessary to select a computing unit and a computational algorithm implementation tailored to the process being carried out. This article examines the performance of various implementations of the commonly used FFT algorithm in terms of execution time and energy consumption on a selected computing unit – the ADSP-SC589 DSP controller. The study examined implementations using the SIMD architecture of the computing unit, as well as implementations using a hardware calculation accelerator built into the device structure. The obtained results suggest significant improvement in signal processing algorithm execution time (around 40 times shorter) of SIMD based implementation as well as minor reduction of power consumption, as well as very high improvement in execution time of hardware accelerator based implementation (around 1000 times shorter) at cost of increased power consumption. The results suggest various implementations significantly differ from each other, and show clear benefits of code optimization based on specific architecture as well as hardware acceleration.

Keywords: DSP, signal processing, edge computing, hardware acceleration.

Introduction

The development of technical facility identification methods regarding optimization of control as well as diagnostics is closely related to progress in evolution of signal acquisition and processing systems and tools.

Simultaneously to the popularization of centralized approach to data management, based on the concept of cloud computing, the interest in edge computing methods is growing. These methods consist of data processing model in which computational activity is shifted towards direct source of data. In technical appliances the source of data is commonly understood to be various types of

sensors. Signals acquired from these sensors, after processing, deliver the information required for conducting control processes or diagnostics.

Edge computing systems are commonly used in machine bearing diagnostics systems, which base their principle of operation on computing and tracking VRMS value directly at the vibration sensor. Such implementations are described by VERSTRAETEN et al. (2019).

The main principle of edge-computing is the reduction of time from the moment of acquisition of information from the sensor and acquiring the result of its' interpretation, as well as reducing the required throughput of communication channels, which connect the sensor with higher layers of the system.

In most cases the condition for effective implementation of edge-computing-based device is its' ability to execute the computational algorithm in given time. The basic limitation for such an approach is the ability of the microcontroller supporting the sensor to execute increasingly complex computational algorithms. Related difficulties are increasing in case of mobile devices, when issues related to energy consumption gain significance. In HIDA et al. (2017) authors illustrate new demands regarding devices used in embedded implementations of IOT systems.

In numerous works authors point out both software and hardware methods of achieving desired effectiveness of given embedded device. However, assessing the performance of microcontroller-based devices in edge-computing applications is ever so important as well as ambiguous. In most cases the choice of optimal solution is a compromise between processing speed and complexity of algorithm, final informational value and energy consumption of given device. The criteria on which this compromise is based are contradictory.

The most important criteria of this compromise are the number of cycles required to complete a task, and the time of single cycle. Core clock frequency based upon these criteria translates into required, specific amount of energy supplied to the device.

Looking at the problem from the perspective of the program structure, it is desirable to have a form in which the total number of machine cycles correlated with the number of basic arithmetic and logical operations reaches a minimum value. Such targeted code optimization based on computational complexity and Big-O notation, as stated by MALA et al. (2022) can reduce program execution time.

The issues of code optimization discussed in the researchers' work refer to both low-level programming, e.g., as stated by MACKOWSKI and NIEZABITOWSKI (2015), and high-level

programming, as per LUO et al. (2009). MACKOWSKI and NIEZABITOWSKI (2015) state, that an example of energy load variation for selected assembler instructions and bus status is given. In LUO et al. (2009), the authors examine the energy consumption of a microprocessor system and implement methods to reduce energy consumption in the programming environment by optimizing the algorithm and program code written in C. Optimization of algorithm, as well as selecting higher compiler code optimization levels provided significant reduction in both execution time and energy consumption. An assessment of the varying impact of code optimization on energy consumption for different microcontrollers, based on statistical analysis using sound statistical methods, is presented by ORTIZ and SANTIAGO (2008). An example of energy consumption analysis related to a specific program code implementation is provided in GUO et al. (2021) regarding encryption algorithms in wireless communications. Obtained results based on instruction-level energy consumption of encryption algorithms under test suggest significantly higher energy consumption of RSA encryption algorithm as opposed to AES256 or SHA256 algorithms.

The desired operation of the device based on the program code is closely related to the proper selection of its hardware layer. The architecture of the controller used is significant for achieving the required processing parameters and maximum energy consumption. The approach to this issue presented in numerous studies includes, among others, such solutions as parallelization of calculations, the use of peripheral devices such as accelerators, and a flexible approach to core clocking.

Contemporary general-purpose microcontroller designs often venture into the realm of signal processing. In computationally demanding applications, dedicated DSP processors still have the upper hand. In both approaches, researchers and manufacturers propose individual improvements. A typical way to increase performance is to parallelize calculations on systems equipped with more than one core. With a properly designed program, such systems can achieve higher performance in some applications. In most cases, microprocessors with a homogeneous architecture between individual cores are used for this purpose. There are also systems offering a heterogeneous set of cores, including those specialized for performing only a part of the total computational tasks.

KUMAR et al (2023) propose an architecture with heterogeneous cores. The proposed computing unit consists of cores operating with the same set of instructions but with different parameters. Test results indicate improved thermal and energy efficiency of the system.

HOMAYOUN (2016) discusses processor architectures designed for use in servers. They propose processors with a heterogeneous architecture – cores with specialized microarchitectures working in conjunction with computing accelerators. One of the requirements for such devices in the context of big data is low energy consumption. Obtained results suggest that big servers deliver higher computational performance compared to little servers but are not as energy efficient. Simultaneously, little servers deliver higher energy performance when processing small data sizes.

Microcontrollers also use specialized circuits – accelerators with narrow functionality, designed for maximum efficiency in terms of both time and energy consumption (OZAKI 2011). In HIDA et al. (2017), the authors examine the energy consumption efficiency of a dynamically reconfigurable computing accelerator compared to a typical RISC microcontroller. The inclusion of support for such a device in the program is ultimately intended to allow some of the computational tasks to be transferred from the main processing unit to the accelerator. ASLAN and YILMAZER-METIN (2022) analyze the energy consumption of GPUs for embedded applications (Nvidia Jetson).

When performing tasks on the accelerator, the main control unit is solely responsible for managing data flow. OZAKI (2011) examines the possibility of using a reconfigurable computing accelerator to transfer part of the calculations from the main processing unit in order to reduce energy consumption.

The methods of powering such systems are evolving, as exemplified by the emergence of research on energy management techniques, as stated by THAKKAR et al. (2020). SOFIANIDIS et al. (2025) investigate energy saving methods in embedded systems, such as Dynamic Voltage and Frequency Scaling. The results suggest that for tasks with specific cycle count required for execution voltage supplied to embedded processor should be as low as possible, and its clock frequency as high as supported at that voltage. However for tasks with specific execution time lowest core frequency at lowest supported voltage should be used to obtain maximum energy savings.

WU et al. (2021) point to the relationship between the processor clock frequency and the power consumption of a popular microcontroller from the Cortex M4 family designed for low-power applications.

The effectiveness of the proposed solutions has been the subject of numerous studies. The authors examine selected methods for reducing energy consumption in microcontroller systems and test the efficiency of the proposed energy-saving strategy on the STM32F407 microcontroller.

Such studies are carried out using different methodologies depending on the objective. GUO and CI (2017) describe approaches involving the use of real computing units with implemented processing algorithms and the simulation of the operation of a given computing unit in an engineering environment.

The measurement-based approach is based on the assumption that a program running on an embedded system controls its hardware layer, which requires electricity to operate. Therefore, measuring the current consumed by the tested device is an intuitive method of assessing the energy efficiency of a given algorithm implementation.

NING et al. (2024) examine the energy consumption of various optimization algorithms on a personal computer using Joulemeter software, which allows the power consumption of a device to be determined. The results obtained formed the basis for conclusions about the efficiency of individual implementations. OSOLINSKYI et al (2023) examines the energy efficiency of a microcontroller on which pattern recognition algorithms are implemented. Energy consumption is determined by measuring the voltage on the power lines and the current on one of the lines. JANKOVIC and DRNDAREVIC (2015) examine the energy consumption of a microcontroller system in various operating modes by measuring the voltage drop across a shunt resistor.

The model approach presented by SHYE et al. (2009) involves analyzing energy consumption data using statistical modeling methods. In the publication, the authors examine the energy consumption of an Android device using an application that logs user activity and propose strategies for reducing energy consumption based on how the device is used. The authors use linear regression modeling. NASH et al (2025) state, that when determining the energy consumption of the system, the authors use linear regression equations to determine the ratio between the individual components that make up the total energy consumption of the system based on the recorded data.

The simulation approach involves determining the energy consumption of a system based on an evaluation of its performance when run on a simulation platform. Such a platform allows a program to be run on a device embedded on another platform (e.g., a personal computer) and its projected energy consumption to be assessed using an appropriate energy consumption model (e.g., a single-instruction energy consumption model). The simulation approach is used in by MACKOWSKI and NIEZABITOWSKI (2015) as well as LUO et al. (2009).

Research methodology

In order to compare the efficiency of the FFT calculation algorithm using three different hardware tools that can be used in the SHARC family DSP system, an experiment was conducted. An experiment was conducted which involved determining the spectrum of a random vibroacoustic signal while measuring the time needed to execute a sequence of instructions and measuring the energy consumed during the execution of this task.

The object of the study was the ADSP-SC589 processor. It is a triple-core processor from the SHARC family manufactured by Analog Devices, based on two cores with SHARC+ architecture and an Arm Cortex A5 core. The SHARC core architecture is designed for efficient floating-point data processing. The processor allows for the implementation of computational algorithms using implementations provided by the manufacturer in the form of functions that can use conventional data processing algorithms, as well as implement them using the SIMD approach and using built-in hardware peripheral computing circuits – FFT accelerators.

SIMD (single instruction, multiple data) is a data processing parallelization technique that involves the simultaneous processing of multiple data items using a single instruction. It is one of the approaches to data processing described by Flynn's taxonomy (FLYNN 1996). In this approach, a single decoded instruction performs arithmetic operations on multiple elements of a data vector. This approach is hardware-dependent, as processors designed to implement this technique (SIMD processors) must have the appropriate architecture, which includes, among other things, vector registers. This processing method is characterized by increased performance compared to the SISD (Single Instruction Single Data) approach. Details of this solution are described by FLYNN (1972) , CYPHER and SANZ (1989), and BHUYAN (1982).

Peripheral accelerators are computing units separate from the microcontroller core, specialized for executing a given computing algorithm or part thereof independently of the main computing unit. The role of the main processor is limited to configuring the unit, providing data, and retrieving results. Due to their specialized nature, accelerators are more efficient than conventional computing units in terms of energy consumption and execution time.

The ADSP SC-589 signal processor is part of the ADSZ SC589 MINI evaluation kit.

The SC-589 is a three-core microcontroller with a heterogeneous set of cores, designed for use in signal processing systems. The microcontroller consists of one ARM Cortex A5 architecture

core, which is a general-purpose core, and two SHARC architecture cores, designed for efficient signal processing operations. All cores have access to the same set of peripherals and can exchange data with each other using shared RAM areas.

Key parameters of ADSP SC-589 processor consist of:

- Maximum core clock frequency of 500 MHz for both ARM Cortex A5 and SHARC cores,
- 32 kB L1 instruction cache and 32 kB L1 data cache for ARM Cortex A5 core,
- Up to 640 kB of L1 SRAM memory for each of the SHARC cores,
- Onboard FFT/IFFT, FIR, IIR, HAE/SINC acceleration,
- Support for ARM NEON architecture extension.

Input data in the form of a stream of acceleration bit values was obtained from an AVS1001HF vibration sensor equipped with an RS485 communication interface.

The measurement of program execution time was implemented in software using microcontroller peripheral devices (timers). The results obtained in this way were processed internally by the tested unit and sent to a data storage unit running on a computer using serial port communication.

Energy consumption measurements were performed using a power supply, a shunt resistor, and an oscilloscope with a measurement logging function. The DC power supply was used as the power source for the evaluation kit containing the tested processor. A shunt resistor was placed on one of the power supply lines of the kit. The voltage drop across this resistor, proportional to the current flowing through it, was recorded using an external oscilloscope with a measurement logging function. At the same time, the voltage value on the power supply lines was recorded and the resulting power consumption of the system was calculated. The recorded data was saved on an external data carrier.

The program was implemented on two cores of the ADSP SC-589 device. Core 0, with ARM CORTEX A5 architecture, was responsible for performing initialization activities in the program, as well as for handling the presentation of results and communication with the external programming environment. Core 1, with SHARC architecture, was used to implement the signal processing algorithm and control measurement activities.

In order to obtain data for the planned comparative analysis, a measurement stand was constructed, the diagram of which is shown in Fig 1.

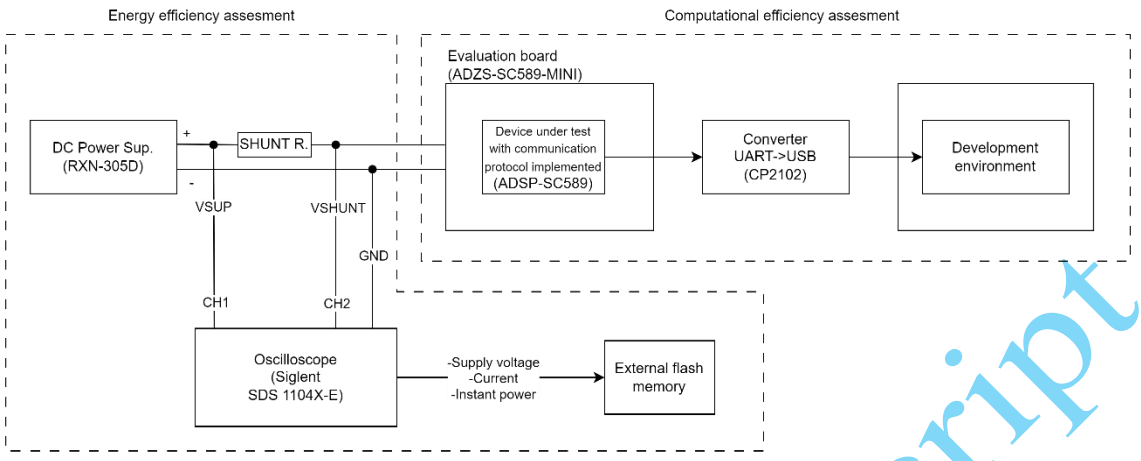

Fig. 1. Diagram of the measurement stand

The measurement stand consists of a part responsible for monitoring parameters related to energy consumption by the tested system, and a part responsible for measuring the performance of algorithm implementations in terms of algorithm execution time.

An experiment was conducted at the measuring station according to the following scenario:

- **Initialization** using core 0, including: variable initialization, device port line configuration, timer configuration, serial port configuration for communication with the sensor.
- **Establishing communication** with the AVS1001HF vibration sensor.

  Phase 1: reading the status message and configuration from the sensor.

  Phase 2: sending the command to start data acquisition

  Phase 3: Next, a command was sent to the acceleration sensor to return the collected sample sequence. The data sequence collected by the sensor was received by the ADSP-SC589 device and stored in an intermediate data buffer. Due to the formatting used by the sensor manufacturers (a stream of 24-bit integers), the data stream had to be adapted for further processing by reformatting it to a steam of 32-bit floating point values.

The obtained floating point values were written to a buffer located in memory accessible to all cores of the DSP device.

Start of execution time measurement by core 1.

Execution of the signal processing algorithms.

End of execution time measurement.

Transferring program execution back to core 0, whose task is to transfer the results to an external data storage unit using a serial port.

Power measurements were performed during the cyclical implementation of the evaluation program. The recorded power consumption values were collected using an oscilloscope and saved on an external data carrier. According to the diagram in Fig 3.

The division of tasks between individual cores of the ADSP system is shown in Fig 2.

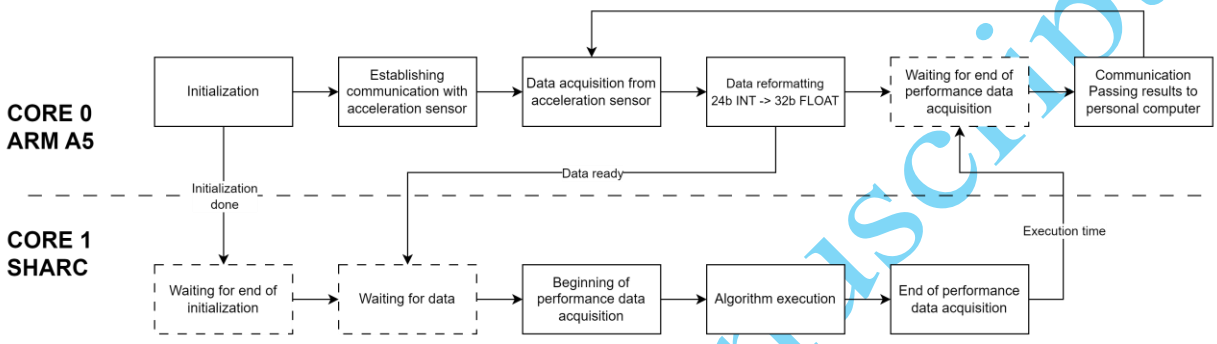

Fig. 2. Distribution of measurement system tasks among individual cores of the ADSP-SC589 system in terms of algorithm execution time measurement

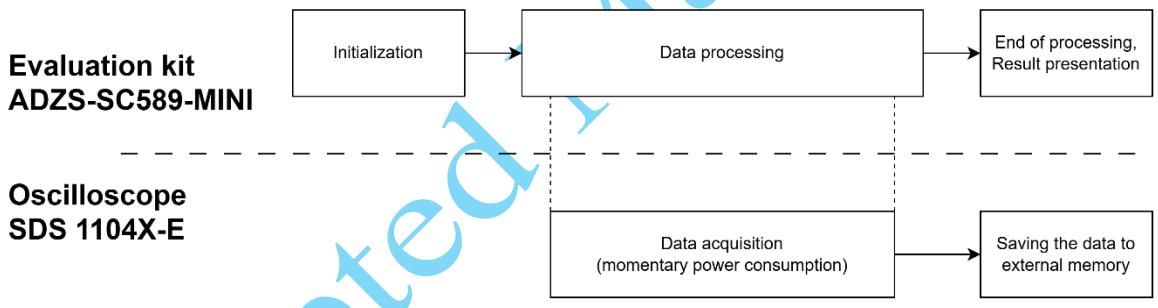

Fig. 3. Power measurement diagram using an external logger

The presented measurement scenario containing implementations of the FFT algorithm on 1024 points was performed three times in different hardware implementations:

1. Implementation not using the SIMD approach – TRANS_RFFT1024, (SA – Simple Algorithm)
2. Implementation using the SIMD approach – FILTERH_RFFT1024, (SIMD)
3. Implementation using a peripheral FFT accelerator – ACCEL_RFFT1024, (ACCEL.)

In order to determine the relationship between energy consumption and the time needed to complete the processing of a given algorithm, both quantities were measured synchronously according to the diagram in Fig 4.

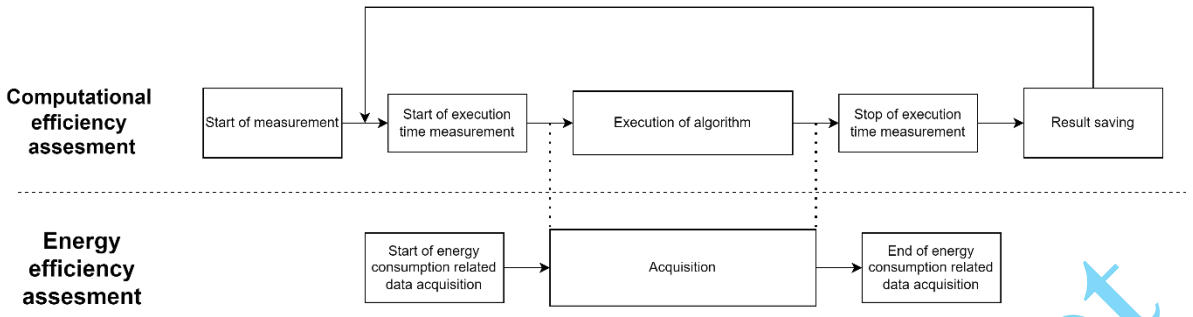

Fig. 4. Diagram of the method for assessing the energy consumption and execution time of individual implementations of the FFT algorithm

Results

Data on time and power consumption during the execution of individual implementations are summarized in Table 1.

Table 1: Summary of parameters for individual implementations of the FFT1024 algorithm.

Implementation	Execution time [us]	Power cons. [W]	Energy per cycle [uJ]
SA	88839,847	1,073	95325,156
SIMD	2203,451	1,002	2207,858
ACCEL	94,327	1,253	118,192

The graphical illustration of the results is presented in figures 5,6 and 7.

Fig. 5. Comparison of execution times for individual implementations
(SA – Simple Algorithm, SIMD – Implementation using SIMD, ACCEL – Implementation using peripheral accelerator).

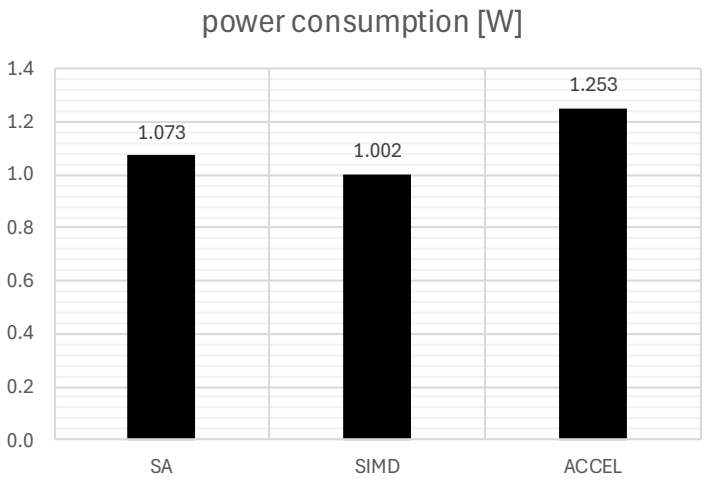

Fig. 6. Summary of the unit energy consumption of various implementations

Table 2: Summary of statistical parameters of energy consumption measurement for various implementations of processing algorithms.

Implementation	Mean power cons. [W]	Variance	Standard deviation
SA	1.073	0.091	0.008
SIMD	1.002	0.095	0.009
ACCEL	1.253	0.096	0.009

The power consumption values of the system during operation were recorded, as well as the average values of the time required to execute the algorithm.

Table 1 contains the values of unit energy consumption during the implementation of the FFT1024 algorithm. The summary of power values is presented in the bar chart in Fig 6.

A summary of the time values for individual algorithms is presented in Fig 5.

In terms of the time needed to execute the algorithm, the SA implementation took the longest – 88839.847us. The SIMD implementation took about 40 times less time. The implementation based on the use of a peripheral computing accelerator (ACCEL) took the least time – 94.327us.

In terms of unit energy consumption, the implementation of the algorithm based on the peripheral computing accelerator (ACCEL) – 1.253W. This is about 20% more energy consumption than in the case of the other implementations considered. The SIMD-based implementation had the lowest energy consumption among the implementations tested – 1.002W.

The energy consumption of the SA implementation was comparable to that of the SIMD implementation.

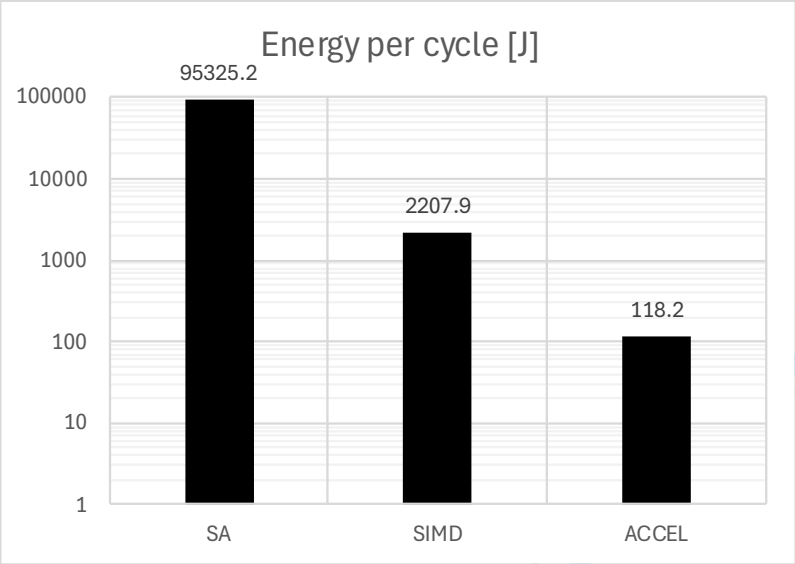

Fig. 7. Summary of energy consumption per single FFT cycle for individual implementations

Based on the data on times and power, the energy required to perform a single FFT cycle for each implementation was determined. The obtained values are presented in Fig 7.

The most efficient implementation in terms of energy consumption per single FFT algorithm cycle is the one based on a peripheral computing accelerator. It achieved energy consumption per cycle approximately 800 times lower than the SA implementation. The SIMD-based implementation was approximately 40 times more efficient than the SA implementation.

The results obtained from the research indicate significant differences between individual implementations of the FFT1024 signal processing algorithm, both in terms of execution time and energy consumption.

The implementation of the FFT algorithm based on the use of a peripheral computing accelerator (ACCEL) showed the highest recorded unit energy consumption by the DSP device during calculations – an average of 1.253W. Among the implementations performed directly on the DSP core, the implementation using the SIMD approach achieved slightly lower energy consumption (1.002W) than the implementation without SIMD (1.073W).

The SIMD-based implementation took approximately 40 times less time per execution than the SA implementation. This indicates a significant reduction in execution time compared to SA. The most time-efficient implementation is the one based on a peripheral computing accelerator. This

suggests that it is appropriate to use an accelerator-based algorithm implementation for time-critical tasks.

Conclusions

The implementation based on a computation accelerator (ACCEL) showed the most favorable energy consumption per single algorithm execution cycle. This suggests the possibility of using accelerator-based computations for energy-critical tasks

The results concerning the energy consumption of the evaluation set when executing the algorithm in individual implementations indicate significant differences in energy consumption between implementations executed exclusively on the DSP core and the implementation based on hardware acceleration.

The results describing the program execution time for individual implementations indicate a significant time gain when using both SIMD-based implementations and implementations using hardware acceleration.

The use of an implementation based on hardware acceleration results in a significant gain in program execution time compared to implementations performing calculations on the DSP core, while increasing energy consumption by approximately 20%.

References

VERSTRAETEN, T., MARULANDA, F.G., PEETERS, C., DAEMS ,P.J., NOWÉ, A., HELSEN, J. 2019. Edge computing for advanced vibration signal processing. Surveillance, Vishno and AVE conferences, INSA-Lyon, Université de Lyon. https://hal.science/hal-02188766v1.

HIDA, I., TAKAMAEDA-YAMAZAKI, S., IKEBE, M., MOTOMURA, M., ASAI, T. 2017. A High Performance and Energy Efficient Microprocessor with a Novel Restricted Dynamically Reconfigurable Accelerator. CS, vol. 08, no. 05, pp. 134–147. doi: 10.4236/cs.2017.85009.

MALA, F.A., ALI, R. 2022. The Big-O of Mathematics and Computer Science. JAMC, vol. 6, no. 1, pp. 1–3. doi: 10.26855/jamc.2022.03.001.

MACKOWSKI, M., NIEZABITOWSKI, M. 2015. Power Consumption Analysis of Microprocessor Unit Based on Software Realization. 2015 20th International Conference on Control Systems and Computer Science, Bucharest, Romania: IEEE, 493–498. doi: 10.1109/CSCS.2015.75.

LUO, G., GUO, B., SHEN, Y., LIAO, H., REN, L. 2009. Analysis and Optimization of Embedded Software Energy Consumption on the Source Code and Algorithm Level. 2009 Fourth International Conference on Embedded and Multimedia Computing, Jeju, Korea (South): IEEE, 1–5. doi: 10.1109/EM-COM.2009.5402965.

ORTIZ, D.A., SANTIAGO, N.G. 2008. Impact of source code optimizations on power consumption of embedded systems. 2008 Joint 6th International IEEE Northeast Workshop on Circuits and Systems and TAISA Conference, Montreal, QC, Canada: IEEE, 133–136. doi: 10.1109/NEWCAS.2008.4606339.

GUO, C., YANG, Y., ZHOU, Y., ZHANG, K., CI, S. 2021. A Quantitative Study of Energy Consumption for Embedded Security. 2021 IEEE Wireless Communications and Networking Conference (WCNC), Nanjing, China: IEEE. 1–5. doi: 10.1109/WCNC49053.2021.9417382.

KUMAR, R., FARKAS, K.I., JOUPPI, N.P., RANGANATHAN, P., TULLSEN, D.M. 2003. Single-ISA heterogeneous multi-core architectures: the potential for processor power reduction. 22nd Digital Avionics Systems Conference. Proceedings (Cat. No.03CH37449), San Diego, CA, USA: IEEE Comput. Soc. 81–92. doi: 10.1109/MICRO.2003.1253185.

HOMAYOUN, H. 2016. Heterogeneous chip multiprocessor architectures for big data applications. Proceedings of the ACM International Conference on Computing Frontiers, Como Italy: ACM. 400–405. doi: 10.1145/2903150.2908078.

OZAKI N. 2011. Cool Mega-Arrays: Ultralow-Power Reconfigurable Accelerator Chips. IEEE Micro, 31(6) 6–18. doi: 10.1109/MM.2011.94.

ASLAN, B., YILMAZER-METIN, A. 2022. A Study on Power and Energy Measurement of NVIDIA Jetson Embedded GPUs Using Built-in Sensor. 2022 7th International Conference on Computer Science and Engineering (UBMK), Diyarbakir, Turkey: IEEE. 1–6. doi: 10.1109/UBMK55850.2022.9919522.

THAKKAR, A., CHAUDHARI, K., SHAH, M. 2020. A Comprehensive Survey on Energy-Efficient Power Management Techniques. Procedia Computer Science, 167, 1189–1199. doi: 10.1016/j.procs.2020.03.432.

SOFIANIDIS, I., KONSTANTAKOS, V., NIKOLAIDIS, S. 2025. Reducing Energy Consumption in Embedded Systems Applications. Technologies, 13(2), 82. doi: 10.3390/technologies13020082.

WU, H., CHEN, C., WENG, K. 2021. An Energy-Efficient Strategy for Microcontrollers. Applied Sciences, 11(6), 2581. doi: 10.3390/app11062581.

GUO, C., CI, S. 2017. A Survey of Energy Consumption Measurement in Embedded Systems. IEEE Access 2017, 5, 10068-10080. doi: 10.1109/ACCESS.2017.2712261.

NING, Z., VANDERSTEEGEN, M., VAN BEECK, K., GOEDEME, T., VANDEWALLE, P. 2024. Power Consumption Benchmark for Embedded AI Interface. International Conferences on Applied Computing.

OSOLINSKYI, O., LIPIANINA-HONCHARENKO, K., KOCHAN, V., SACHENKO, A., ZAHORODNIA, D. 2023. Energy Consumption of Methods for Pattern Recognition using Microcontrollers. IJC, 502–508. doi: 10.47839/ijc.22.4.3358.

JANKOVIC, S.P., DRNDAREVIC, V.R. 2015. Microcontroller power consumption measurement based on PSoC. 2015 23rd Telecommunications Forum Telfor (TELFOR), Belgrade, Serbia: IEEE, 673–676. doi: 10.1109/TELFOR.2015.7377557.

SHYE, A., SCHOLBROCK, B., MEMIK., G. 2009. Into the Wild: Studying Real User Activity Patterns to Guide Power Optimizations for Mobile Architectures". MICRO'09.

NASH, D.C., MARTIN, T.L., HA, D.S., HSIAO, M.S. 2005 Towards an Intrusion Detection System for Battery Exhaustion Attacks on Mobile Computing Devices. Third IEEE International Conference on Pervasive Computing and Communications Workshops, Kauai Island, HI, USA: IEEE, 2005, 141–145. doi: 10.1109/PERCOMW.2005.86.

FLYNN, M.J. 1972. Some Computer Organizations and Their Effectiveness. IEEE Transactions on Computers 1972, C-21(9). doi: 10.1109/TC.1972.5009071.

CYPHER, R., SANZ J.L.C. 1989. SIMD architectures and algorithms for image processing and computer vision. IEEE Transactions on Acoustics, Speech, and Signal Processing 1989, 37, 12). doi: 10.1109/29.45558.

BHUYAN L.N., AGRAWAL D.P. 1982. Applications of SIMD computers in signal processing. AFIPS 1982. doi: 10.1109/AFIPS.1982.17.

JANKOVIC P.S., DRNDAREVIC V.R. 2015. Microcontroller Power Consumption Measurement Based on PSoC. 23rd Telecommunications Forum Telfor (TELFOR). doi: 10.1109/TELFOR.2015.7377557.

FLYNN M.J. Very High-Speed Computing Systems. 1966. PROCEEDINGS OF THE IEEE, 54, 12, DECEMBER. doi: 10.1109%2FPROC.1966.5273.