Mirosław KOZIOŁ, Janusz KACZMAREK, Ryszard RYBSKI
Uniwersytet Zielonogórski
Instytut Metrologii, Elektroniki i Informatyki

METHOD FOR QUICK CALCULATION OF SINE SAMPLES IN THE POLYPHASE SINUSOIDAL DIGITAL SIGNAL GENERATORS

In some applications of sinusoidal digital signal generators, it is necessary to change the parameters of the generated signal very often, and therefore quickly calculate the sine samples. The chapter presents a method of sine samples calculation with a constant angle increment and any initial phase with minimal calls of standard trigonometric function. The saved time was determined, which is obtained when the described method is utilized in the sine waveform computation, both in the absence of hardware support for calculations on double-precision floating-point numbers, as well as when such hardware support is provided.

METODA SZYBKIEGO OBLICZANIA PRÓBEK FUNKCJI SINUS W WIELOFAZOWYCH CYFROWYCH GENERATORACH SYGNAŁÓW SINUSOIDALNYCH

W niektórych zastosowaniach generatorów konieczna jest bardzo częsta zmiana parametrów generowanego sygnału, a w związku z tym szybkie obliczenie próbek sinusa. W rozdziale przedstawiono metodę wyznaczania próbek sinusa ze stałym przyrostem kąta i dowolnej fazie początkowej przy minimalnym stosowaniu standardowej funkcji trygonometrycznej. Określono zysk czasu, jaki uzyskuje się stosując opisaną metodę przy wyznaczaniu wzorca przebiegu sinusoidalnego zarówno przy braku wsparcia sprzętowego dla obliczeń na liczbach zmiennoprzecinkowych podwójnej precyzji, jak również gdy takie wsparcie jest.

1. INTRODUCTION

Sources of digitally synthesized sinusoidal voltages, called sinusoidal digital signal generators, have been used for many years in the alternating current measurement techniques owning to the fact that they can generate amplitude, phase and frequency stable signals, and provide high resolution amplitude, phase and frequency setting. Working in a complex measurement system they also allow clock synchronization among all the components of the system. Thanks to it e.g. data acquisition devices can perform coherent sampling to avoid leakage and FFT windowing. One of the areas in which the use of sinusoidal digital signal generators has significantly influenced the development of new measurement devices and systems is an impedance measurement [1-6].

The impedance measurement is important not only in electrical engineering and electronics. Technological progress related to the availability of better and better electronic systems, especially high-resolution digital-to-analog (DAC) and analog-to-digital (ADC) converters, contributes to increasing the accuracy of impedance measurements, which in turn increases the use of impedance measurement in disciplines such as medicine [7] and biology [8], as well as to agricultural product quality monitoring [9]. On the other hand, this progress contributes to the creation of new materials and objects that require the use of these impedance methods [10]. Therefore, measuring the impedance or detecting its small changes plays an important role in material testing related to the latest technologies [11, 12].

In the mentioned areas of impedance measurement, in many cases, the time needed to perform those measurements plays a significant role. It is determined, regardless of the subject of the research, by the applied measurement methods and their system implementation. In measurement systems using sinusoidal digital signal generators, especially polyphase, where it is necessary to regulate the frequency, amplitude and phase of the generated voltages during the measuring process with a high resolution, the time needed to generate voltages with the given parameters is important. Accurate impedance measurement systems and power calibrators are good examples. In case of three-phase power calibrators, parameters in 3 voltage and 3 current channels are regulated in each cycle of the generation and measurement process (control loop cycle). Digital impedance bridges designed for the comparison of four-terminal-pair impedance standards use 6-7 channel sinusoidal digital signal generators which parameters are adjusted during a relatively complex balancing process. The time needed to perform the measurement can be reduced by using e.g. an appropriate balancing algorithm or by using an unbalanced bridge [13]. Nevertheless, in

each of these two approaches, the calculation time of the values of the sinusoidal function samples, most often representing one period of the generated signal, is of significant importance. The calculation time depends on the number of channels for which the parameters of the generated signal are changed in a given measurement cycle. At a constant rate of sample transfer to DACs, the reduction of the signal frequency is achieved by increasing the number of samples per period. Therefore, the signal frequency also affects the calculation time. The use of a constant sample rate results from the requirements for low-pass filters at the outputs of DACs. This is of particular importance in precise systems for generating sinusoidal waveforms using digital methods.

2. POLYPHASE SINUSOIDAL DIGITAL SIGNAL GENERATORS – CONSTRUCTION AND PRINCIPLE OF OPERATION

Sinusoidal digital signal generators usually consist of four functional blocks, as shown in Figure 1. A numerically controlled oscillator (NCO) provides a representation of the sinusoidal signal discrete in time and value in a form of digital codes that are synchronously and sequentially sent to the next block, i.e. DAC. The set of codes, hereinafter referred to as a sine waveform pattern, is determined on the basis of the discrete sine function values (the sine samples), taking into account the given value of amplitude and initial phase.

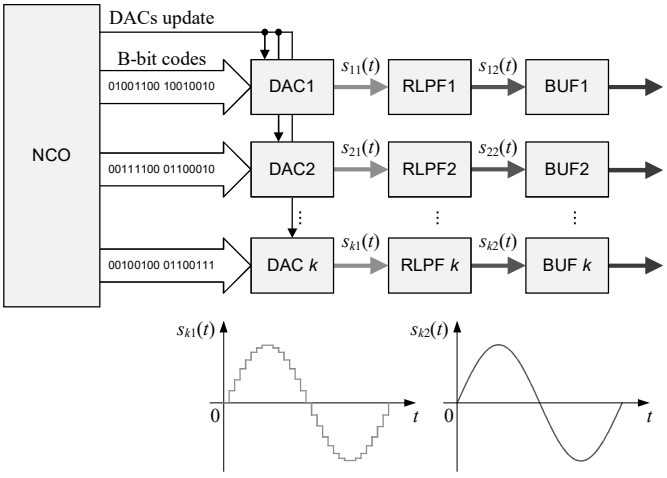

Fig. 1. Functional blocks of the polyphase digital signal generator and shape of signals at the outputs of DAC and RLPF blocks

Rys. 1. Bloki funkcjonalne wielofazowego cyfrowego generatora i kształt sygnałów na wyjściach bloków DAC i RLPF

There are several ways to implement the NCO. The three most frequently used are listed in [14]. The simplest approach is based on a digital counter addressing a look-up table implemented in ROM memory, where the set of digital sine samples is stored. Replacing the digital counter with the registered adder forming so called phase accumulator leads to a second DDS-based NCO implementation method [15]. The last one is based on the sine samples calculation in software [16-18].

The task of DAC is to convert the codes into a voltage signal, which is the approximation of the sine function by the staircase waveform. From the spectral point of view, the signal at the output of DAC contains not only the fundamental frequency of the generated sinusoidal signal but also its images around the clock frequency and integer multiples thereof. Due to the fact that the images are not desired, DAC is followed by a reconstruction low-pass filter (RLPF) that removes all the spectral components resulting from the staircase approximation of the desired sine waveform.

To ensure adequate current efficiency, a buffer (BUF) is placed at the end of the entire processing chain. In case of polyphase generators, the last three blocks (DAC, filter and buffer) are duplicated allowing simultaneous and synchronous generation of multiple signals.

3. THE METHOD OF CALCULATING THE SAMPLES OF THE SINE FUNCTION

In the digital impedance bridges, the balancing process requires frequent changes in amplitude and phase. If the NCO is implemented in software, the amplitude and/or phase changes require new calculation of sine samples. Since trigonometric calculations are time consuming, to reduce computational burden an idea, presented in [19], of calculation of sines and cosines of consecutive angles can be used. This approach reduces the calls upon standard trigonometric functions based on the following two trigonometric identities:

$$\sin(x+y)=\sin x\cdot\cos y+\cos x\cdot\sin y, \tag{1}$$

$$\cos(x+y)=\cos x\cdot\cos y-\sin x\cdot\sin y. \tag{2}$$

For the purpose of this method, the identities (1) and (2) are written as follows

$$\sin(\alpha+n\alpha)=\sin\alpha\cdot\cos(n\alpha)+\cos\alpha\cdot\sin(n\alpha), \tag{3}$$

$$\cos(\alpha + n\alpha) = \cos\alpha \cdot \cos(n\alpha) - \sin\alpha \cdot \sin(n\alpha), \tag{4}$$

where α is the assumed angle increment, which depends on the number N of samples per period of the generated sine waveform according to the following relationship

$$\alpha = \frac{2\pi}{N}, \tag{5}$$

and n takes the values of successive natural numbers from 0 to N–1. In the application under consideration, the identity (4) has an auxiliary function due to the occurrence of the cosine function of the integral multiple of α in the equation (3).

The method described above allows to determine the value of the sine function for integer multiples of α only for initial phase with the value of zero, which in the case of the discussed application in generators is a significant limitation. However, a slight modification of the presented method allows to overcome this disadvantage.

Denoting the initial phase of the sine by φ, the computation of the sine samples $s[n]$ with a minimal use of a standard trigonometric function call can be performed on the basis of the following equation

$$s[n] = \sin(\varphi + n\alpha) = \sin\varphi \cdot \cos(n\alpha) + \cos\varphi \cdot \sin(n\alpha), \tag{6}$$

taking into account the identities (3) and (4), which in this case play the role of auxiliary formulas for determining the value of the sine and cosine of successive integer multiples of the angle α required in the equation (6).

The procedure for calculating samples of one sine period is carried out in three steps. In the first one, the values of the sine and cosine should be computed using standard trigonometric function call for the angle equal to the initial phase φ and for the assumed angle increment α.

$$A = \sin\varphi \tag{7}$$

$$B = \cos\varphi \tag{8}$$

$$C = \sin\alpha \tag{9}$$

$$D = \cos\alpha \tag{10}$$

The value of A is also the value of the first sample of the sine, i.e. $s[0]$.

In the second step, the second sample of the sine is computed according to the equation (6) for $n = 1$. Because the values of trigonometric functions required in this formula were determined in the first step, so the sine sample $s[1]$ can be computed according to the following equation

$$s[1] = \sin(\varphi + n\alpha) = A \cdot D + B \cdot C . \tag{11}$$

In the third step, the remaining values of the sine samples with the initial phase φ and successive integer increments of the angle α are computed. First, two new designations C_{prev} and D_{prev} are introduced with initial values of C and D, respectively. Then, for successive integer values of n from 2 to N–1 according to the equations (3) and (4) new values of the sine and cosine are computed, i.e.:

$$C_{\text{new}} = \sin(\alpha + n\alpha) = C \cdot D_{\text{prev}} + D \cdot C_{\text{prev}}, \tag{12}$$

$$D_{\text{new}} = \cos(\alpha + n\alpha) = D \cdot D_{\text{prev}} - C \cdot C_{\text{prev}}, \tag{13}$$

which allows to compute the value of the next sine sample

$$s[n] = \sin(\varphi + n\alpha) = A \cdot D_{\text{new}} + B \cdot C_{\text{new}} . \tag{14}$$

At the end, before changing the value of n, the following substitutions are made

$$C_{\text{prev}} = C_{\text{new}} , \tag{15}$$

$$D_{\text{prev}} = D_{\text{new}} . \tag{16}$$

As it can be seen, the standard trigonometric functions are used only in the first step of the calculations. Calculating the value of the sine sample in the second step (for $n = 1$) requires only 2 additions and 1 multiplication, and the calculation of each next sample (for $n = 2$ .. N–1) requires 6 multiplications and 3 additions.

4. EXPERIMENTAL RESULTS

4.1. Validation of the DAC codes calculation

The validation of the DAC codes $CODE_{\mathrm{CLC}}$ obtained on the basis of sine samples calculated using the presented method was performed by comparing them with the reference codes $CODE_{\mathrm{REF}}$ according to the formula

$$CODE_{\mathrm{ERR}} = CODE_{\mathrm{REF}} - CODE_{CLC} \,. \tag{17}$$

The reference codes were calculated on the basis of standard sine function available in the C language and were implemented here using double-precision floating-point numbers.

The presented method of calculating sine samples can be implemented with the use of single- or double-precision floating-point numbers. In the first step, single-precision variables were used in its implementation. As the proposed method of calculating the sinus samples is intended to speed up the computation of sine waveform pattern in the generator presented in [20, 21], where 20-bit DACs with an output voltage range of ±10 V are utilized, such values were assumed for these two parameters when carrying out the analyses.

Figure 2 shows the differences among codes computed for 7 different initial phases, $N = 25000$ and amplitude of 9,9 V. Regardless of the initial phase, the difference in the codes, in general, increases for the consecutive sample numbers, reaching the value of 0 for those sample numbers that fall at zeros of the sine function.

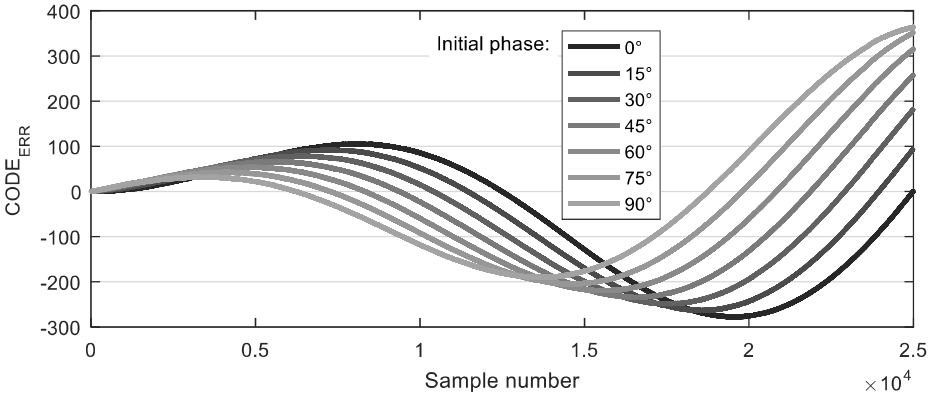

Fig. 2. Difference among codes the of the sine waveform pattern depending on the initial phase

Rys. 2. Różnica między kodami we wzorcu przebiegu sinusoidalnego w zależności od fazy początkowej

The validation of the method was also performed for each combination of 10 different amplitudes and 6 values of the number of samples *N* per period of the generated waveform pattern. The obtained results are shown in the Figure 3 and 4.

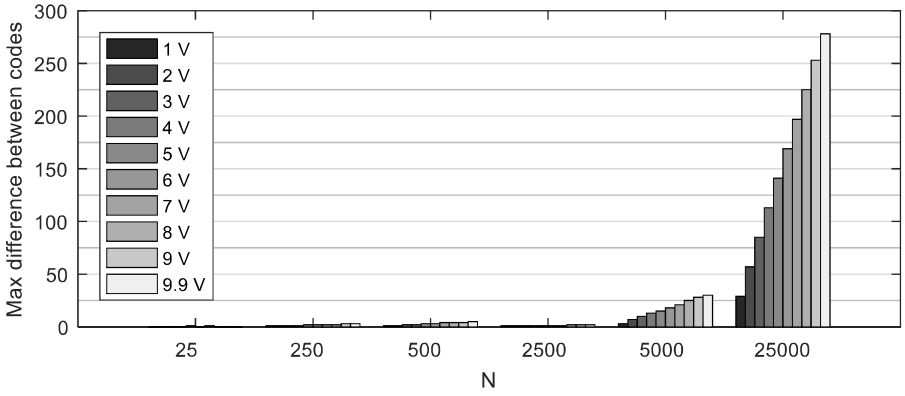

Fig. 3. Maximum difference among codes depending on the number of samples per period of the sine waveform pattern and the amplitude
Rys. 3. Maksymalna różnica pomiędzy kodami we wzorcu przebiegu sinusoidalnego w zależności od liczby próbek na okres wzorca i amplitudy

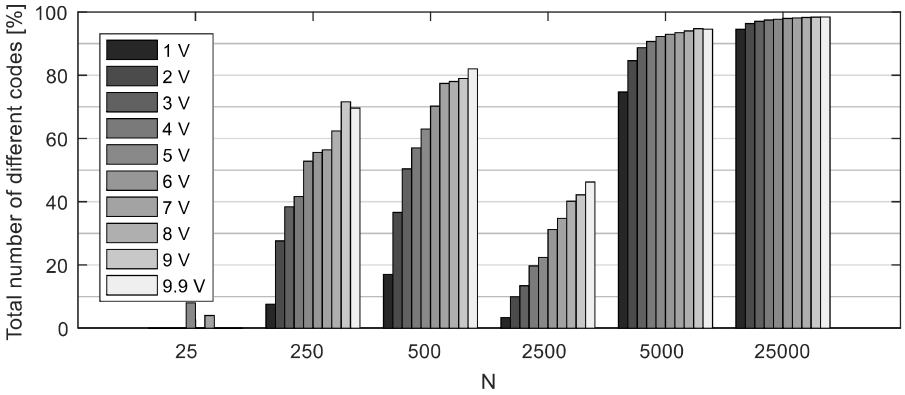

Fig. 4. Total number of different codes depending on the number of samples per period of the sine waveform pattern and the amplitude
Rys. 4. Całkowita liczba kodów o innej wartości we wzorcu przebiegu sinusoidalnego w zależności od liczby próbek na okres wzorca i amplitudy

The graph in Figure 3 shows the maximum error $CODE_{ERR}$ obtained for a given amplitude and number of samples N. As it can be seen, for N lower than or equal to 2500, the differences are insignificant and the codes differ by no more than 5 regardless of the amplitude. However, when N is higher, the differences are much greater. The graph in Figure 4 shows the maximum percentage of different codes in the waveform pattern for a given amplitude and number of samples N. The greater the N, the greater the percentage of different codes, and for 25000 samples, regardless of the amplitude, it reaches almost 95%.

However, if the presented method of computing sine samples is implemented using double-precision floating-point numbers, then regardless of the amplitude, initial phase and number of samples per signal period, the difference in the codes is exactly 0. Therefore, further tests were carried out using the implementation of the sine sample calculation procedure based on the double-precision floating-point numbers.

4.2. Speed analysis of the sine waveform pattern calculation

The microcontroller STM32F746, which implements, among others, the NCO functionality in the generator described in [20, 21], is based on the ARM Cortex-M7 processor. Figure 5 illustrates the system architecture in the STM32F7 series of microcontrollers [22].

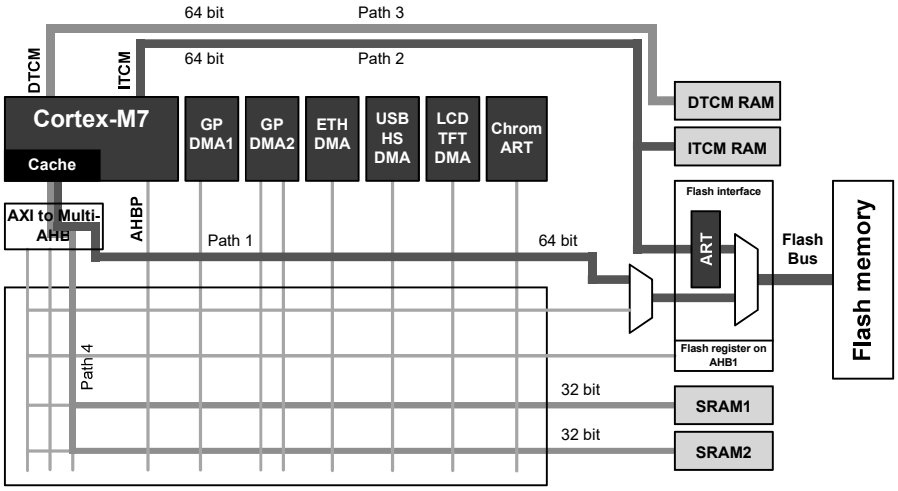

Fig. 5. The system architecture in the STM32F7 series of microcontrollers
Rys. 5. Architektura mikrokontrolerów serii STM32F7

In all the devices the Flash memory is accessible via an AXI/AHB bridge with enabled or disabled instruction cache (Path 1 in Figure 5) or via a ITCM bus (Path 2). As the embedded Flash memory is slow compared to the processor core, the Adaptive Real-Time accelerator (ART) is provided for a Flash memory access on the ITCM interface to unleash the Cortex-M7 core performance and allow 0-wait code execution. For the same reason the instruction cache should be enabled when the Flash memory is accessible via the AXI/AHB bridge.

In turn, the variables involved in computation of the sine samples may be allocated in DTCM-RAM mapped on the DTCM interface (Path 3 in Figure 5) and accessed at a maximum processor clock speed without latency or in SRAM memory (Path 4). To speed up the access to the SRAM memory the data cache can be enabled. The various access paths presented here were taken into account when running the tests.

The utilized microcontroller has hardware support only for the execution of calculations on single-precision floating-point numbers. As shown above, for the proposed method of calculating sine samples this is not applicable. Nevertheless, it was checked how much the time of determining the sine waveform pattern would be shortened only by applying the proposed method of sine samples calculation without hardware support for calculations on double-precision floating-point numbers.

The Cortex-M7 processor contains a 32-bit free running counter that counts CPU clock cycles. The counter is a part of the Debug Watch and Trace module and it was used to measure the execution time of code responsible for the calculation the sine waveform pattern codes for DAC. The time measurements were carried out for $N = 500$, which corresponds to the calculation of the 1-kHz sine waveform pattern (the samples are sent to DACs with a frequency of 500 kHz).

The saved time ST expressed in percentage obtained as a result of the application of the described method of sine sample calculation was determined from the following relationship

$$ST = \left(1 - \frac{t_{\mathrm{CLC}}}{t_{\mathrm{REF}}}\right) \cdot 100\%, \tag{18}$$

where t_{CLC} is the time of determining the sine waveform patter by calculating the sine samples using the proposed method, and t_{REF} is the calculation time of the sine waveform pattern by calculating the sine samples using the standard trigonometric function call. The obtained results are shown in the Figure 6. Regardless of the interface on which the processor accessed the program or data

memory, the use of the proposed sine sample calculation method leads to approximately 75% reduction in computation time. This translates into a 4-fold acceleration of calculations.

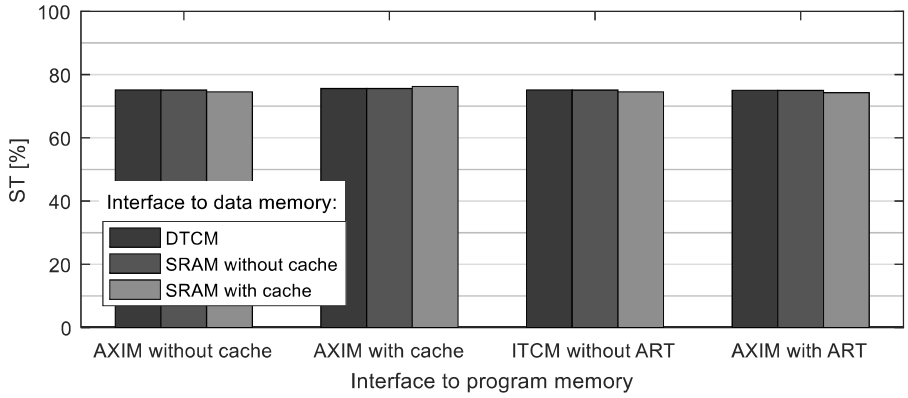

Fig. 6. Saved time in percentage of the sine waveform pattern calculation depending on the way the processor without the hardware support for double-precision floating-point number calculations accesses the data and program memory
Rys. 6. Zaoszczędzony czas w procentach wyznaczania wzorca przebiegu sinusoidalnego w zależności od sposobu dostępu procesora bez sprzętowego wsparcia dla obliczeń na liczbach zmiennoprzecinkowych podwójnej precyzji do pamięci danych i programu

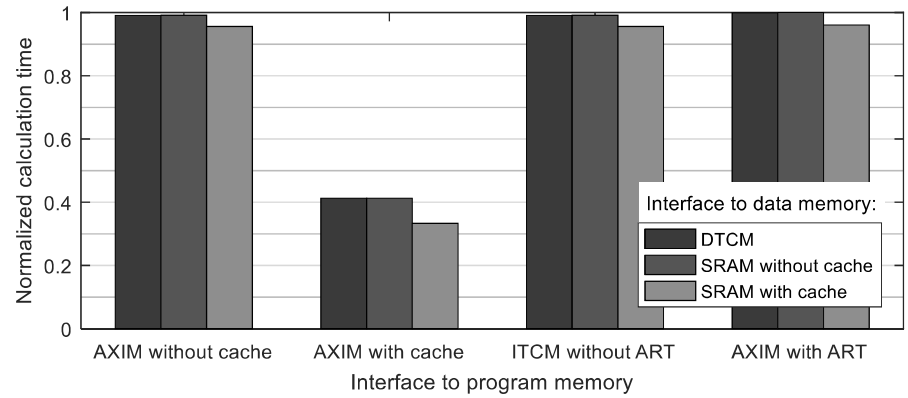

Fig. 7. Normalized calculation time of the sine waveform pattern depending on the way the processor accesses the data and program memory
Rys. 7. Znormalizowany czas wyznaczania wzorca przebiegu sinusoidalnego w zależności od sposobu dostępu procesora do pamięci danych i programu

Considering only the times of determining the sine waveform pattern performed with the use of the proposed method of calculating the sine samples, the best results are achieved when the processor accesses the program memory

via the AXIM interface with enabled instruction cache. The type of data memory and the way of accessing it do not make any major differences in times. This is shown in the graph in Figure 7. The normalized time on the Y axis was determined in relation to the longest time obtained during the measurements that took place when the program memory was accessed via the ITCM interface with the ART accelerator enabled, and the data was stored in the SRAM memory with enabled data cache.

To check how hardware support for double-precision floating-point calculations can affect the determination time of the sine waveform pattern, the procedure using the proposed method for sine samples calculation was run on the STM32F767 microcontroller, which has such support. The device was clocked with the same frequency. The only difference from the previous device is in the cache size, which is 4 times greater (16 KB). The graph in Figure 8 shows the saved time in percentage compared to the time obtained with the previous microcontroller. As it can be seen, the hardware support for double-precision floating-point calculations in most cases leads to 95% reduction in computation time. It means that the sine waveform pattern can be calculated 20 times faster than without this acceleration.

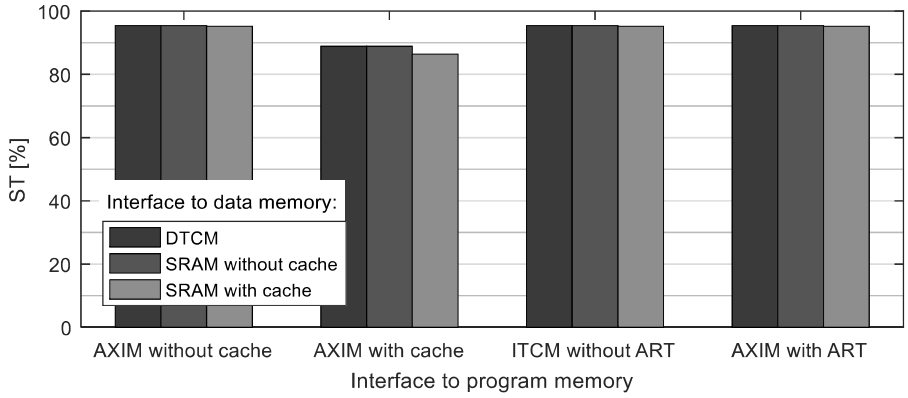

Fig. 8. Saved time in percentage of the sine waveform pattern calculation depending on the way the processor with the hardware support for double-precision floating-point number calculations accesses the data and program memory

Rys. 8. Zaoszczędzony czas w procentach wyznaczania wzorca przebiegu sinusoidalnego w zależności od sposobu dostępu procesora ze sprzętowym wsparciem dla obliczeń na liczbach zmiennoprzecinkowych podwójnej precyzji do pamięci danych i programu

5. CONCLUSIONS

The chapter presents a method of sine samples calculation with a constant angle increment and any initial phase with minimal calls of standard trigonometric function. The implementation of this method with the use of single-precision floating-point numbers in the generator software in order to accelerate the sine waveform pattern determination showed its imperfection. Therefore, the tests were carried out using the implementation on the double-precision floating-point numbers.

The obtained results show that even in the absence of hardware support for calculations on double-precision floating-point numbers, the use of the presented method causes a 4-fold acceleration of the sine waveform pattern determination in relation to the case where the sine samples are calculated by calling the standard trigonometric function. When such support is available, another 20-fold acceleration of calculations is achieved.

REFERENCES

1. Callegaro L.: Electrical Impedance: Principles, Measurement, and Applications (Series in Sensors) 1st Ed., CRC Press, 2012.
2. Overney F., Jeanneret B.: Impedance bridges: from Wheatstone to Jo-sephson, Metrologia, vol. 55, no. 5, Oct. 2018.
3. Ortolano M. et al.: An international comparison of phase angle standards between the novel impedance bridges of CMI, INRIM and METAS, Metrologia, vol. 55, no. 4, Aug. 2018.
4. Kučera J., Kováč J.: A Reconfigurable Four Terminal-Pair Digitally Assisted and Fully Digital Impedance Ratio Bridge, IEEE Trans. on Instrumentation and Measurement, vol. 67, no 5, May 2018.
5. Kozioł M., Kaczmarek J., Rybski R.: Characterization of PXI-based generators for impedance measurement setups, IEEE Trans. Instrumentation and Measurement, vol. 68, 2019.
6. Kampik M., Musioł K.: Investigations of the high-performance source of digitally synthesized sinusoidal voltage for primary impedance metrology, Measurement, vol. 168, Jan. 2021, available online 3 August 2020.
7. Addabbo T., et al.: On the Suitability of Low-Cost Compact Instrumentation for Blood Impedance Measurement, IEEE Trans. on Instrumentation and Measurement, Vol. 68, No. 7, Jul. 2019.
8. Tan C., Liu S., Jia J., Dong F.: A Wideband Electrical Impedance Tomography System Based on Sensitive Bioimpedance Spectrum Bandwidth, IEEE Trans. on Instrumentation and Measurement, Vol. 69, No. 1, Jan. 2020.
9. Cabrera-López J.-J., Velasco-Medina J.: Structure Approach and Impedance Spectroscopy Microsystem for Fractional-Order Electrical Characterization of

Vegetable Tissues, IEEE Trans. on Instrumentation and Measurement, Vol. 69, No. 2, Feb. 2020.
10. Hoja J., Lentka G.: Method Using Square-Pulse Excitation for High-Impedance Spectroscopy of Anticorrosion Coatings, IEEE Trans. on Instrumentation and Measurement, Vol. 60, No. 12, Dec. 2015.
11. Guha A., Patra A.: Online Estimation of the Electrochemical Impedance Spectrum and Remaining Useful Life of Lithium-Ion Batteries, IEEE Trans. on Instrumentation and Measurement, Vol. 67, No. 8, Aug. 2018.
12. Marraci M., Tellini B., Catelani M, Ciani L.: Ultracapacitor Degradation State Diagnosis via Electrochemical Impedance Spectroscopy, IEEE Trans. on Instrumentation and Measurement, Vol. 64, No. 7, Jul. 2015.
13. Rybski R., Kaczmarek J, Kontorski K.: Impedance Comparison using Unbalance Bridge with Digital Sine Wave Voltage Sources. IEEE Trans. on Instrumentation and Measurement, Vol. 64, No. 12, Dec. 2015.
14. Popek G., Kampik M.: Low-Spur Numerically Controlled Oscillator Using Taylor Series Approximation, XI International PhD Workshop OWD 2009, 17-20 October 2009.
15. Bojdoł G., Kampik M.: Numerically controlled oscillator with reduced effect of phase accumulator truncation on accuracy of generated sine samples, Pomiary Automatyka Kontrola, nr 7-8, 2006.
16. TMS320C62x Algorithm: Sine Wave Generation, Texas Instruments Application Report, SPRA708, Nov. 2000.
17. Generation of Sine Wave Using a TMS320C54x Digital Signal Processor, Texas Instruments Application Report, SPRA819, Jul. 2004.
18. Augustyn J.: An Algorithm for Frequency Estimation of Sinusoidal Signal, Metrology and Measurement System, Vol. IX, No. 4, 2002.
19. Lyons R.G.: Understanding Digital Signal Processing, Pearson Education, 2011.
20. Kozioł M., Kaczmarek J., Rybski R.: High-performance two-phase sine wave generator for impedance bridges, XXI IMEKO Word Congress "Measurement and Research in Industry", August 30 – September 4, 2015, Prague, Czech Republic.
21. Kozioł M., Kaczmarek J., Rybski R., Kučera J.: A two-phase sine wave generator dedicated for impedance comparison systems, Przegląd Elektrotechniczny, R. 93, NR 8/2017.
22. STM32F7 Series system architecture and performance, Application note AN4667, STMicroelecronics, Rev. 4, Feb. 2017.