Marcin CEGIELSKI

Technical University of Lodz

Speed analyse of two step algorithms of trigonometric transformations on multi-core processors

Abstract. *The two-stage trigonometric transformations algorithms have full symmetry calculations for each stage of the algorithm. Such algorithm may be subjected to any decomposition allowing to split the process of the calculations into any number of processes, which can be implemented independently within one step of the algorithm. Additionally, a single step of algorithm may depend on the size of the data and the associated number of arithmetic operations, which implementation may depend on available hardware resources. In the article the results of the computations experiments for multi-core processors are presented and compared.*

Streszczenie. *Dwuetapowe algorytmy przekształceń trygonometrycznych posiadają pełną symetrię obliczeń dla poszczególnych bloków algorytmu. Algorytm taki może być poddany dowolnej dekompozycji pozwalając na rozdzielenie procesu obliczeń na dowolną liczbę procesów, które mogą być realizowane niezależnie w obrębie jednego kroku algorytmu. Dodatkowo pojedynczy krok algorytmu może być uzależniony od wielkości danych i związanych z nim liczby operacji arytmetycznych, których realizacja może być uzależniona od dostępnych zasobów sprzętowych. W artykule zaprezentowano i porównano wyniki szybkości algorytmu otrzymane dla procesorów wielordzeniowych. (**Analiza szybkości dwuetapowych algorytmów przekształceń trygonometrycznych dla procesorów wielordzeniowych**)*

Keywords: DCT, FCT, trigonometric transformation, multi-core processor.
Słowa kluczowe: DCT, FCT, przekształcenia trygonometryczne, processory wielordzeniowe.

Introduction

The two-stage trigonometric transformations algorithms are based on the calculations divided into two parts, in which the operations of the addition and subtraction are performed, as well as, at one stage, the operations of the multiplication. Since the algorithm graph resembles the Fourier transform graph, it is possible to make such a decomposition of the graph which enables to extract a group of elements on which calculations are performed in the same way. By appropriate substitution of addresses assigned to input data and particular steps the corresponding values for the cosine and sine for multiplication operations, we can achieve full symmetry the calculations for each step of the algorithm. Such algorithm may be subjected to any decomposition allowing to split the process of the calculations into any number of processes, which can be implemented independently within one step of the algorithm. Additionally, a single step of algorithm may depend on the size of the data and the associated number of arithmetic operations, which implementation may depend on available hardware resources.

The method of the construction of the universal homogeneous structures of fast algorithms was proposed by Yatsymirskyy in [4]. The rules of building algorithms are based on the idea of sum division into two sections. These sections are regrouped and the same transforms are displaced into complementary ones. After the composition N – point transform are replaced into two N/2 – point transforms. This operation is recursively performed until possibilities of realizations are finished. The formula of the decomposition N-point transform into two N/2-point transforms is shown in formulas 1 and 2, for k=0, 1, 2, …, N/2-1

$$(1)\quad C_N^{IV}(k) = C''(k)C_{4N}^{(2k+1)} + C'\left(\frac{N}{2} - k - 1\right)S_{4N}^{(2k+1)}$$

$$(2)\quad C_N^{IV}(N-k-1) = C\left(\frac{N}{2} - k - 1\right)C_{4N}^{(2k+1)} - C(k)C_{4N}^{(2k+1)}$$

where C transforms are in formulas 3 and 4.

$$(3)\quad C'(k) = C_{n/2}^{IV}[a(n)]$$

$$(4)\quad C''(k) = C_{n/2}^{IV}[b(n)]$$

Finished series a(n) and b(n) are in 5,6.

$$(5)\quad a(n) = x(2n) + x(2n+1)$$

$$(6)\quad b(n) = (-1)^n(x(2n) - x(2n+1))$$

Example graph for four point computational blocks is shown in Figure 1.

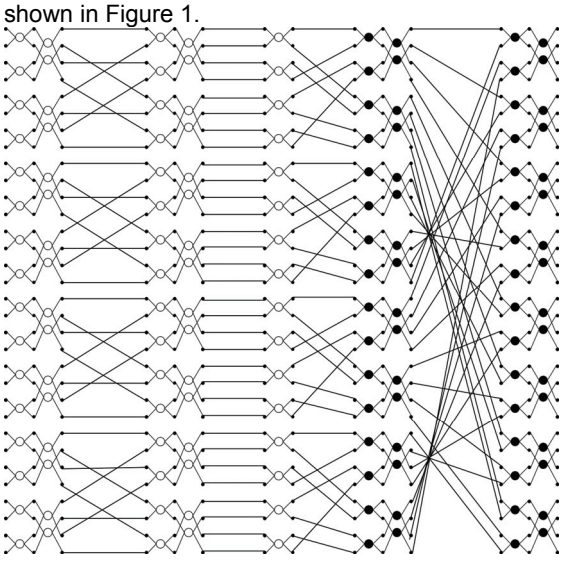


Fig.1. Graph FCT-IV 32-point with division into 4-point blocks

Implementation of the algorithm

The implementation of the block processing technique allows to reduce the number of the indirect references to the operating memory of a computer. As the speed of downloading data to and from computer's memory is much slower than the numerical calculations in numerical coprocessor, the application of the method grouping butterfly calculating in blocks of 4-point rate increase the speed of the calculations of about 10% in comparison to the classical 2-point butterfly algorithm used in PC-class processor (Intel, AMD). The increase in the speed of the calculation is dependent on a particular processor architecture. 64-bit processors of PCs class have been enriched by additional fast internal registers which allow to

increase the size of the calculation block. Multi-core and multi level system of processor cache memory can be expected to further affect the increased speed of multipoint computing of trigonometric transformations.

The structure of multi-core processor is presented in figure 2. We can use 4 cores for parallel computation. 8 point block is computed in each core independently. Communications between cores and memory is accelerated by first level of high speed cache.

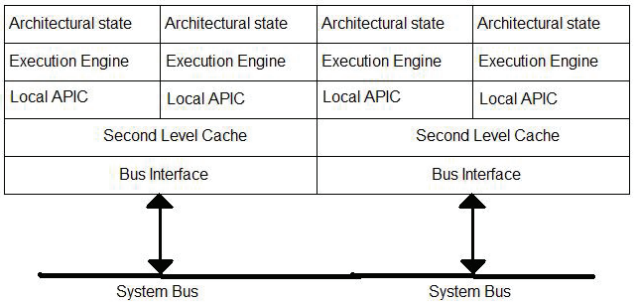


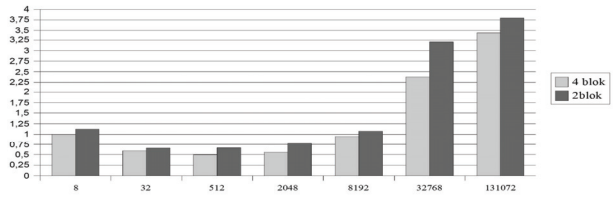
Fig.2. An example of a multi-core processor

It is advisable to compare the speed of the calculation algorithm on multi-core processors using parallel computing for different sizes of trigonometric transformations.

Presentation of experimental results

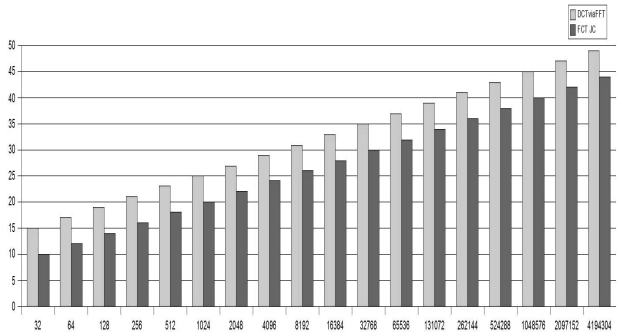
The increase of the speed of calculation associated with increasing of the block calculation rate from 2 to 4 points is presented in Table 1.

Table 1. Computation time FCT-IV algorithms of blocks of 2 and 4-point



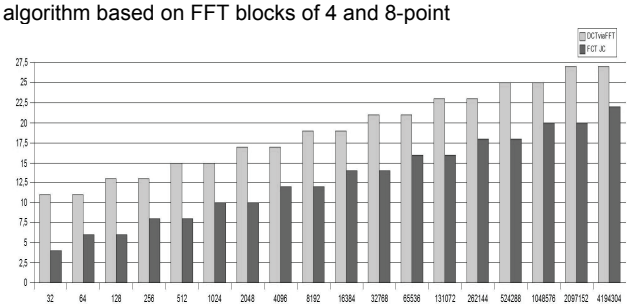
Another important criterion for comparison is the number of operations to download and save the data to an external cache memory type. It should be noted that the execution of operations in the computation block uses different number of core registers. In algorithms based on FFT coefficients are complex numbers and in the two-step algorithms only real. This is due to the fact that for any complex number we need to allocate two registers floating. This means that the FCT algorithm with 4 point block can be compared to FFT algorithm based on a 2 point computational block. The result of this comparison between the FCT for block 4 and DCT based on 2 point FFT is shown in Table 2.

Table 2. Number of in/out two-step algorithms DCT with 4 point blocks and DCT algorithm via FFT with 2 point blocks.



The maximum possible increase in the computational block ia64 processors is 8 points. The outcome of the changes in calculation speed is presented in Table 3.

Table 3. Number of in/out two-step algorithms DCT and DCT algorithm based on FFT blocks of 4 and 8-point



The formula of maximum time computation of parallel FCT-IV is in 7.

$$T_{max} = 6\frac{N}{P_{io}}\log_2(N)\,T_{io} + \frac{N}{P_{OB2}}\log_2(N)\,T_{OB2} + \frac{N}{P_{OB}}\log_2(N)\,T_{OB} \quad (7)$$

where P_{io} - number of possible concurrent read or write operation, a T_{io} - time of singular input/output operations , P_{OB} - number of computations cores, T_{OB} - time of computations in the core.

Conclusion

Symmetrical design and scalability of the algorithms based on the FFT transformation allows to use computer systems equipped with multi-core processors (*Intel Core Class* or *AMD Phemon*). The basic computational block can be implemented to any processor core with its universal registers which allows to calculate 8-point cosine transform without indirect references to the cache or memory. Calculations of more points transform is mainly within the lowest level cache, and references to further coefficients can be reduced to minimum.

REFERENCES

[1] Wolf W., High-Performance Embedded Computing: Architectures, Applications, and Methodologies, *Elsevier* 2007.

[2] Hsiao S.F., Shiue W.R., Tseng J.M., [34] A cost efficient fully-pipelinable architecture for DCT/IDCT, *IEEE Transactions on Consumer Electronics*, 45 (2001), No. 3, 515-525.

[3] Nikara J., Application-Specific Parallel Structures for Discrete Cosine Transform and Variable Length Decoding, *Tampere, Finland* 2004.

[4] Yatsymirskyy M., Shifted in the Time and Frequency domains Cosine and Sine Transforms Fast Algorithms with Homogeneous Structure, *Radioelectronics and Communications Systems,* [52] Allerton Press. Inc, New York Radioelektronika.

[5] *Cegielski M.,* Parallel algorithm of FCT type IV, Proceedings of the 7th International workshop “Computational Problems of Electrical Engineering”, *august 2006, Odessa.*

***Author**: dr inż. Marcin Cegielski, Politechnika Łódzka, Instytut Informatyki, ul. Wólczańska 213, 90-965 Łódź, E-mail:* marcin.cegielski@p.lodz.pl.