Journal of Multidisciplinary Engineering Science and Technology (JMEST)
ISSN: 2458-9403
Vol. 7 Issue 9, September - 2020

Algorithm and Program for Parallel Calculating of Haar Fast Transform in Dual-Core Special Processors

Hakimjon Zaynidinov
Tashkent University of Information Technologies
Tashkent, Uzbekistan
e-mail: tet2001@rambler.ru

Sanjarbek Ibragimov
Andijan Machine Building Institute
Andijan, Uzbekistan
e-mail: sanjari07@yahoo.com

Ibrohimjon Yusupov
Tashkent University of Information Technologies
Tashkent, Uzbekistan
e-mail: Ibrohimbek.211_10@mail.ru

G'ayratbek Tojiboyev
Andijan State University
Andijan, Uzbekistan
e-mail: g_tojiboyev@andmi.uz

Abstract — In this article, parallel algorithms have been developed for digital signal processing on piecewise-polynomial basis of Haar which is based on Analog Devices' Blackfin ADSP-BF561 dual-core processors. Based on these algorithms, developing of dual-core parallel computing program on VisualDSP++ has been considered. Functional descriptions of the Blackfin ADSP-BF561 dual-core processor architecture are given. The architecture and components of the processor core have been described. The VsiualDSP++ application project represents the structure of connecting cores to each other. In the piecewise-polynomial basis of Haar, the results of the time spent on digital processing have been given, to the array that is obtained by analytically on ADSP-BF533 single-core processor and ADSP-BF561 dual-core processors. The times spent on each core were compared and the acceleration coefficient was determined.

Keywords— Blackfin processors family; ADSP-BF561 processors; dual-core processors; processor architecture; digital signal processing; Haar basis function; parallel algorithm of Haar fast transform; acceleration coefficient;

I. INTRODUCTION

Today, digital signal processing technology is commonly being used in practice and with the help of it, it is increasing work efficiency. Digital signal processors and digital signal processing methods are used to solve problems in communications, radio engineering, electronics, acoustics, and seismology, television and control systems. Digital signal processors are manufactured by three major companies: Analog Devices, Freescale and Texas Instruments. These companies are also developing a multi-core signal processor designed to increase data processing capacity and processing speed.

A multi-core processor is a single computing component called a core that has two or more independent computing blocks that reads and executes program commands. Today, multi-core architecture is being used in almost all processors. The use of multi-core processors in digital signal processors, such as universal processors, helps to increase performance more than single-core processors. The ADSP-BF561 and ADSP-BF60x processors from the Blackfin processor family of Analog Devices are a new generation of dual-core 16-32-bit microprocessors. Blackfin Processors combine a 32-bit instruction set specific to RISC processors based on an MSA (Micro Signal Architecture) architecture developed in collaboration with Intel, and a two-block 16-bit MAC duplicate-add signal processing function. This allows Blackfin processors to perform better in signal processing and program performance management, and in many cases eliminates the need for separate identical processors. This ability significantly simplifies both hardware and software project implementation tasks. Also, products from the Blackfin Processors family reduce power consumption by up to 0.8 Volts. Such a combination of high efficiency and low power is very important now and in the future for wireless, mobile, signal processing applications on portable devices connected to the Internet.

Due to the development of technical means, the demand for software for parallel computing of signals based on multi-processor and multi-core processors is growing. In this article, we will look at the parallel calculation of basic functions with Haar fast transform algorithms, which are widely used in solving digital signal processing problems based on Blackfin ADSP-BF561 dual-core processors from Analog Devices.

II. ARCHITECTURE

A. *Architecture of ADSP-BF561 Processor.*

Blackfin ADSP-BF561 processors are a symmetrical dual-core special processor. The ADSP-BF561 processor uses a hierarchical three-level memory model. The first level L1 memory operates at the clock frequency of the core, but the memory capacity is not very large. Each Blackfin core has its own 100 Kbytes of L1 memory and it includes:

- SRAM / cache 16 K bytes of command memory
- SRAM 16 K byte command memory
- SRAM / cache 32 K bytes of data memory
- SRAM 32 K bytes of data memory
- SRAM scratchpad 4 K bytes of RAM

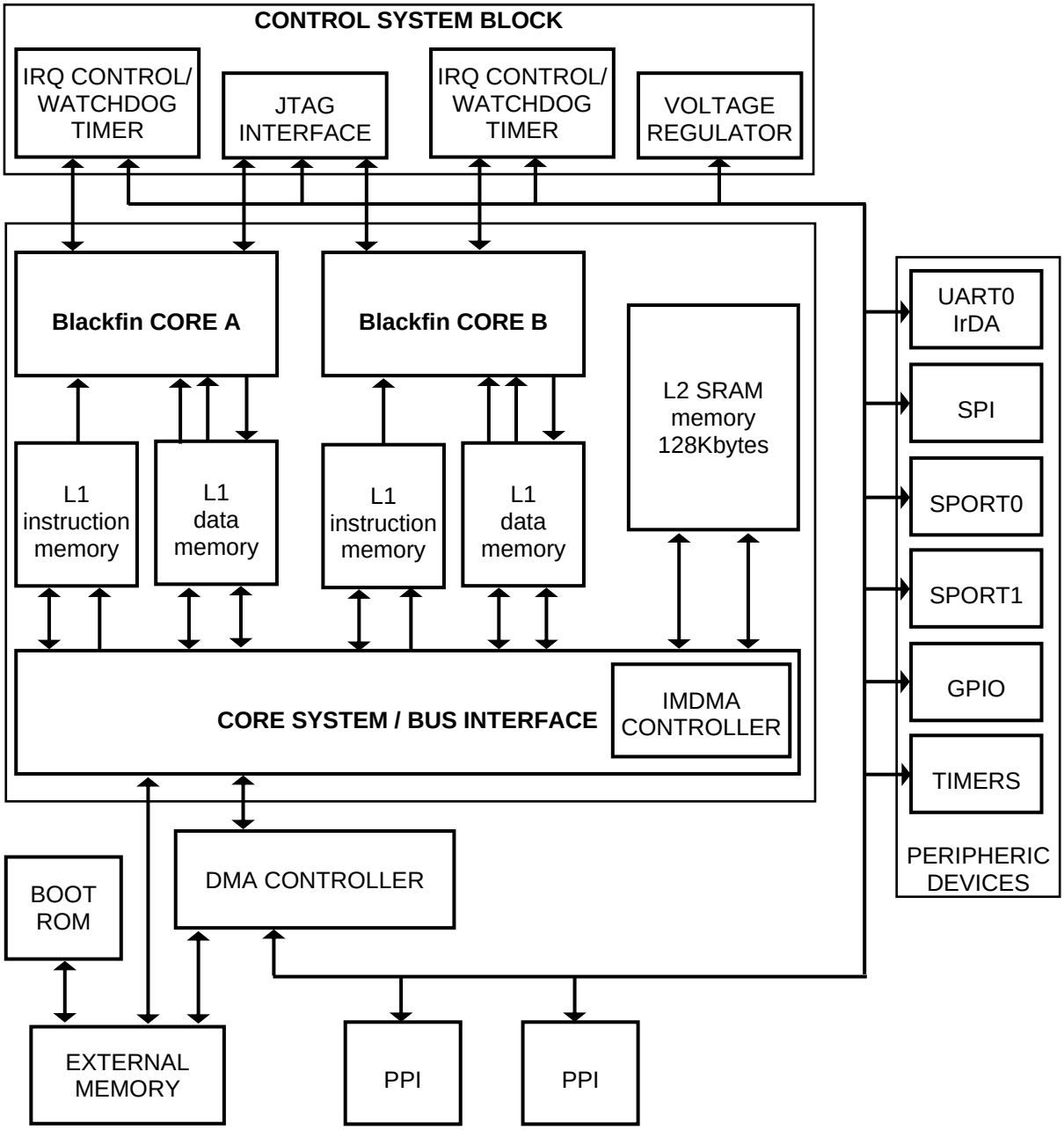

Figure 1. Functional block diagram of ADSP-BF561 processor

The secondary L2 SRAM is integrated with a 128 Kbyte memory core. L2 memory can store common commands and data of both cores. To optimize data exchange between L1 and L2 memories, the processor architecture includes four-channel DMA controllers of special internal memory.

The third-level L3 memory in the hierarchical memory model of the Blackfin processor is external memory. The external memory area can represent four banks of SDRAM from 16 to 512 MB and four asynchronous (ROM, SRAM, EEPROM, flash) memory of 64 MB each.

The processor architecture provides three operating modes: user mode, Supervisor mode, and Emulation mode. In user mode, access to the lower part of the system resources are limited in order to ensure a secure environment of the software. Access to kernel and system resources is not restricted in Supervisor and Emulation modes.

Blackfin processor commands are optimized so that the most commonly used commands are represented by 16-bit codes. Digital Signal Processing (DSP) commands are encoded with 32-bit codes as multifunction commands. The internal bus system and computing units allow each processor core to execute multiple commands in a single cycle, which increases the code density.

A. Processor Core Architecture.

Each core of the Blackfin ADSP-BF561 processor has two 16-bit multipliers, two 40-bit batteries, two 40-bit arithmetic logic devices (ALDs), four 8-bit video ALDs, and one 40-bit drive. Computing units process 8-, 16-, or 32-bit data from a registry file (Figure 2).

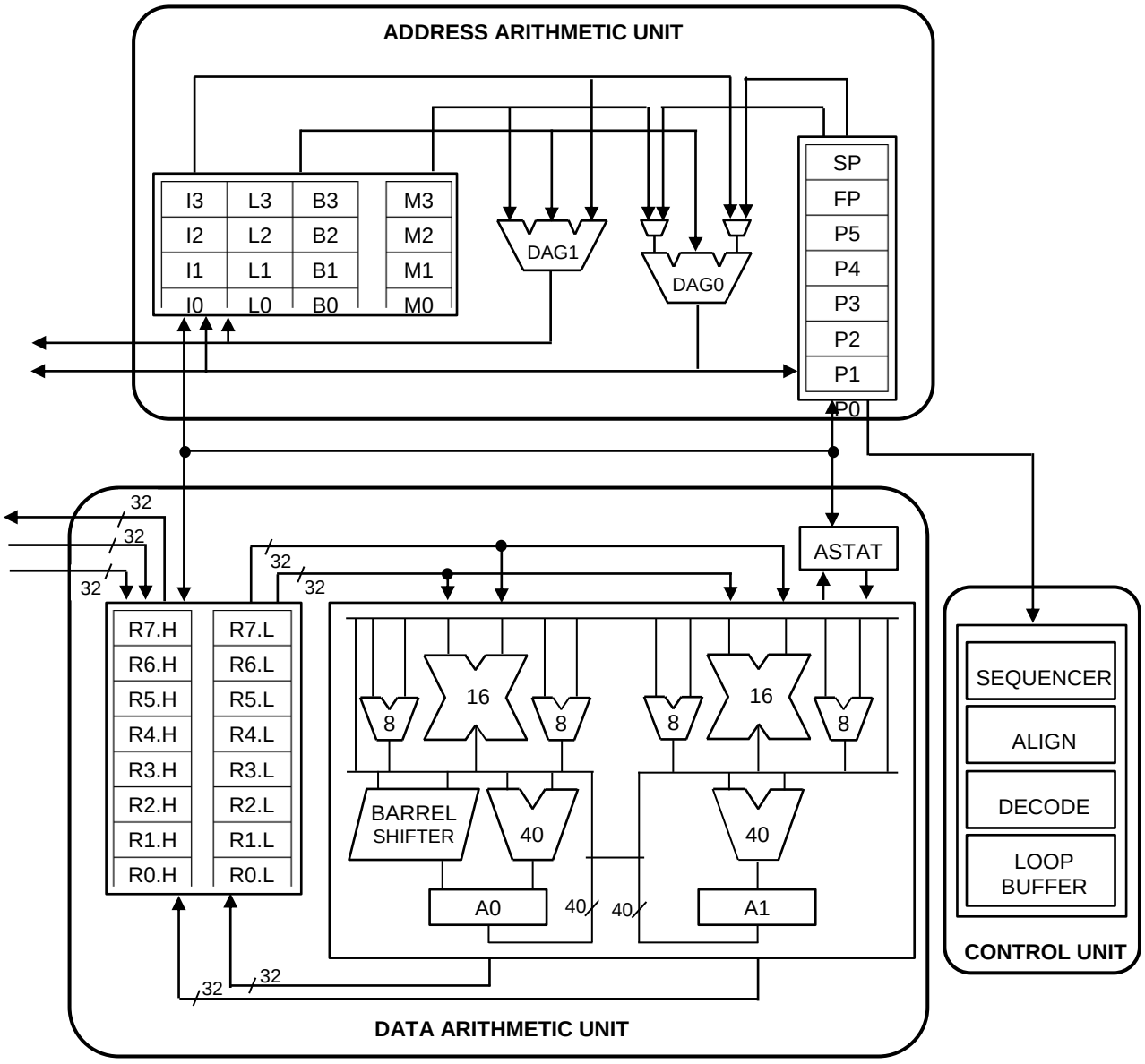

Figure 2. Core architecture of ADSP-BF561 processors

The computational registry file contains eight 32-bit registers. When performing computations with data from 16-bit operands, registry files work like 16 16-bit registers. For computational operations, all operands are taken from a multi-branch registry file and from constant fields.

Each MAC can multiply 16 to 16 bits per clock and can sum up the results to 40 bits. Signal and non-sign formats can be used.

ALD performs the usual set of arithmetic and logic operations on 16 or 32-bit data. It includes many special commands to speed up various signal processing tasks. For some commands, two 16-bit ALD operations can be performed simultaneously as a pair of registers. Four operations of 16 bits can be performed using both ALDs.

Because Blackfin ADSP-BF561 processors contain 2 identical cores, intensive computing applications can distribute computational processes between cores. In

this case, the computational processes of the program code running on both cores will be different or the same. Processed data differs when the program code is the same in both cores. For example, in digital signal data processing, the first core processes half of the data and the second core processes the remaining half of the data.

III. PROGRAMMING DUAL-CORE PROCESSORS ON THE VISUALDSP++ PLATFORM

There are three different types of software design for dual-core Blackfin processors on the VisualDSP++ platform:

- Single-core applications. In this project, only core A is used, while core B is not active.

- One application per core. In this project, each bar core is seen as a separate processor. VisualDSP++ creates a .dxe file for a specific core in the project. Resource sharing is managed by programmers.

- One application across both cores. This creates a common program for the two cores of the project hierarchy. The allocation of resources is controlled by the binding tool.

We use the Single Application / Dual Core type of software project creation for dual-core Blackfin ADSP-BF561 special processors. This type of standard program includes five project hierarchies built as a single program. This is the most efficient approach to ADSP-BF561 programming because it allows all common memory areas to be used efficiently by both cores. To prevent duplication, common code and common data can be stored in shared memory.

Under the high-level executable project of the project hierarchy, there are four sub-projects: A core, B core, L2 internal memory, and external memory. These small projects fall into the DSP Library category. The subprojects create separate files called corea.dlb, coreb.dlb, sml2.dlb and sml3.dlb. Because the application is divided into separate libraries, it is easier to organize the part of the program that is located in a particular kernel or in a specific shared memory.

The software application is divided into three components: two separate cores and a shared memory. The software project creates three main components in a single connection process. This process allows the cores to directly access the code and data in the shared memory, which allows the cores to use a common function or data element.

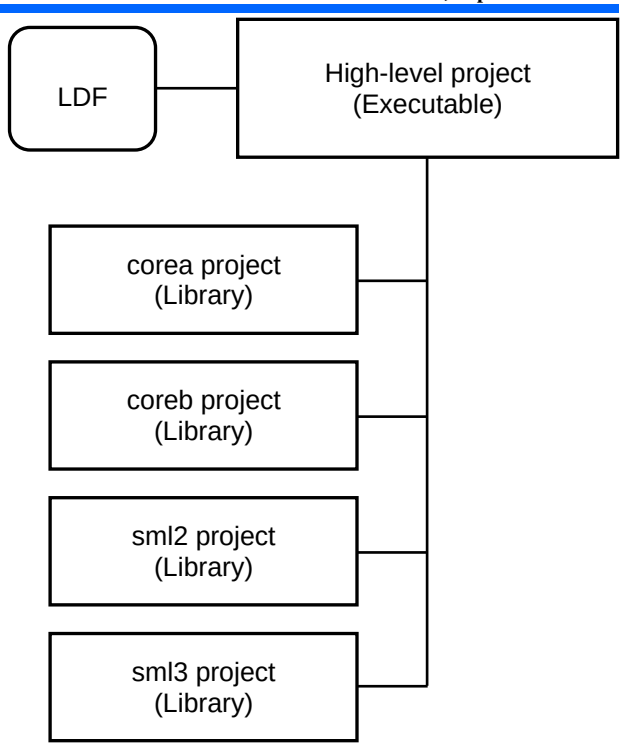

Figure 3. Project hierarchy

A. Multi-Core Connections.

The Single Application / Dual Core approach uses advanced linking tools to resolve interactions between cores and shared memory. Each kernel is defined by a PROCESSOR directive, and together is defined by two areas of memory (internal L2 memory and external memory) and one COMMON_MEMORY (shared memory) directive.

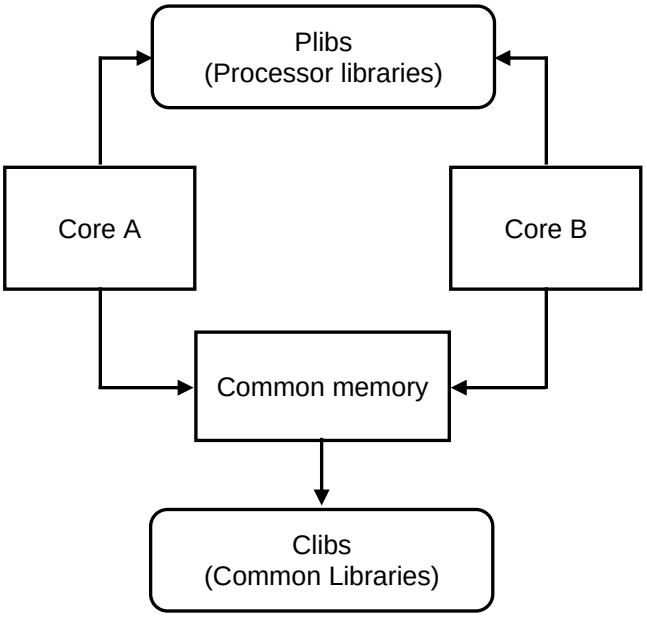

Figure 4. Dual core connection

Both the PROCESSOR directive and the COMMON_MEMORY field can communicate with libraries as shown in Figure 2. Plibs libraries are represented directly according to PROCESSOR instructions. The objects of these libraries are private and are represented in the private memory of the A core or B core. Libraries specified as CLib are represented in the COMMON_MEMORY field. The objects of these libraries are stored in shared memory and can be accessed by both cores. If external data is linked using these libraries, this data is represented in the COMMON_MEMORY field and can be shared between core A and core B.

Common code and data can be represented in the sml2 project in the L2 internal memory or in the sml3 project in the external memory.

In order to share and use data elements between two cores, it is first necessary to identify the common data elements that are used together. The file attribute in the program module is determined by the value MustShare, i.e.

#pragma file_attr ("sharing = MustShare")

General information elements are declared to be variable with the word volatile servant.

IV. ALGORITHM FOR PARALLEL CALCULATION OF HAAR FAST TRANSFORM.

We use Haar Fast transform (HFT) algorithms for digital processing of signals on special dual-core processors. The number of addition and subtraction operations required in the HFT algorithm is 2 (N-1). Here is the number of N-array elements.

Figure 5. shows a graph of Haar fast transform along Andrews. Adding continuous lines in this graph corresponds to the practice of dividing broken lines. The input signals are represented by $X(0), X(1),\ldots, X(N-1)$ and the result is represented by $C(0), C(1),\ldots, C(N-1)$.

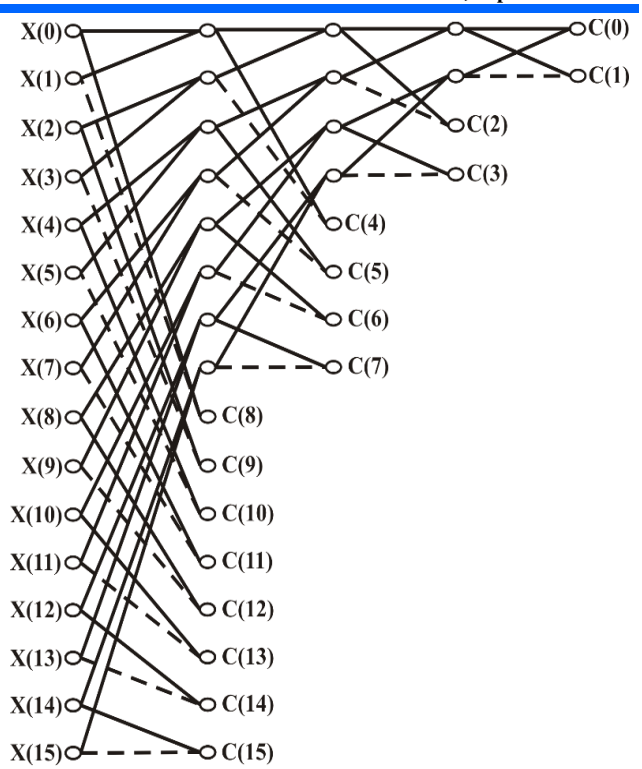

Figure 5. A quick transform graph on Haara Andrews

This graph shows Haar fast transform method on Andrews for a case where N = 16. According to it, in each iteration, the array X is formed from the sum of successive pairs of the previous array X, and the array S is equal to the difference of the successive pairs of the array X.

$$X(i) = X(2i) + X(2i+1) \quad (1)$$

$$C\left(i + \frac{N}{2^j}\right) = X(2i) - X(2i+1) \quad (2)$$

The sum of the Xs calculated in the last iteration

$$C(0) = X(0) + X(1) \quad (3)$$

is equal to here $i = 0\ldots\frac{N}{2} - 1$, $j = 1\ldots k$. The number of iterations is determined by the formula $k = \log_2 N$.

The algorithm of the parallel computational process in the two-core special processor of the above graph is as follows.

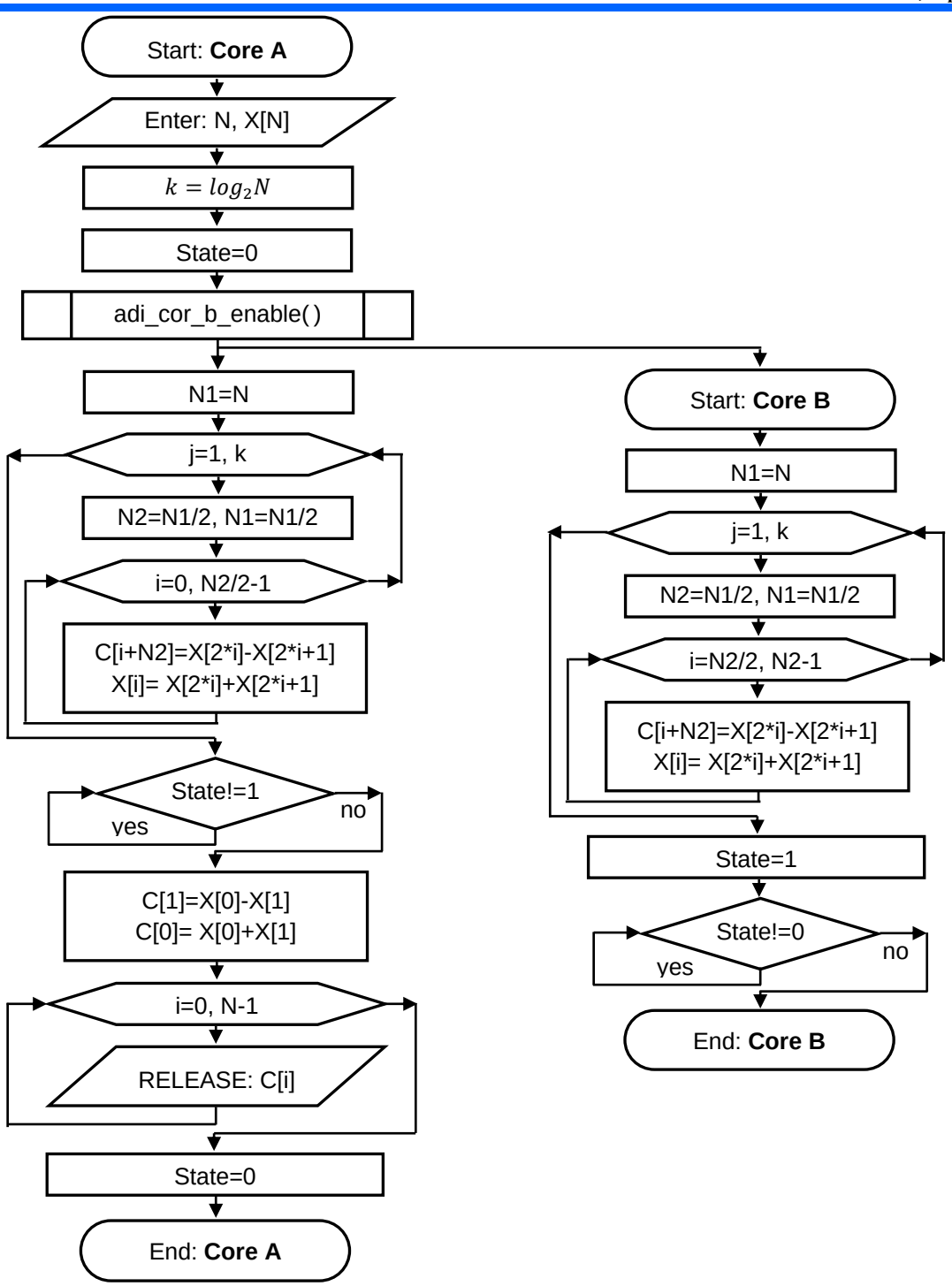

Figure 6. Block diagram of the parallel computational algorithm of Haar fast transform on Andrews on a special dual-core processor Blackfin ADSP-BF561.

According to this algorithm, the processes of input and output of array values are performed sequentially in core A. The computational processes of the HFT are performed in parallel by calculating the first half of the array elements in the A core and the other half in the B core, which is equally distributed in the A core and the B core. Initially, the program code in core A is run and core B is activated using the adi_core_b_anable() function.

An algorithm for parallel computing of HFTs on dual-core processors using fragment-polynomial bases is shown in Figure 7.

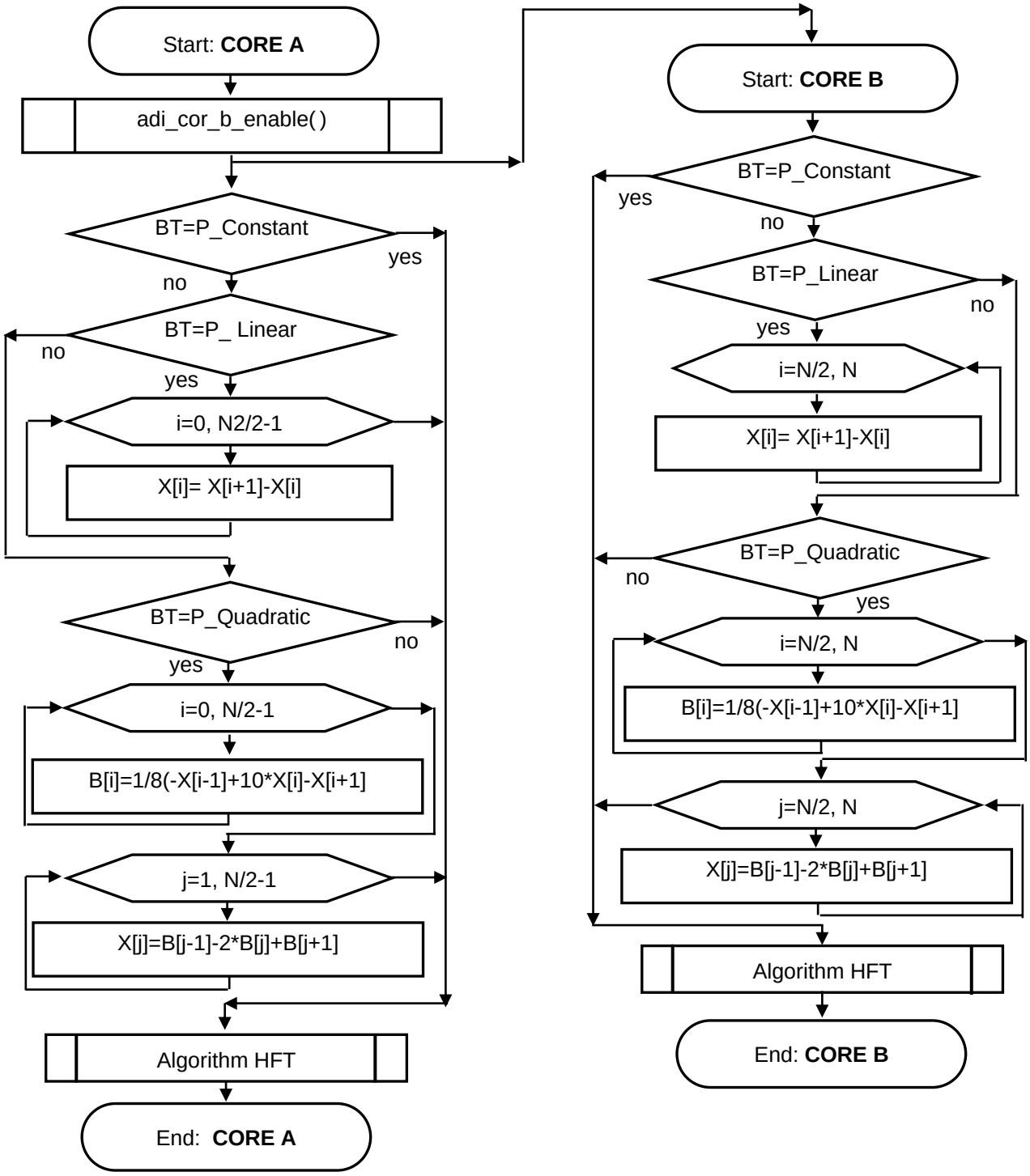

Figure 7. Block diagram of the algorithm for parallel calculation of piecewise-polynomial basis in a special dual-core processor. (BT-Base type, P_Constant-Piecewise constant base, P_Linear- Piecewise linear base, P_Quadratic- Piecewise quadratic base)

V. RESULTS

On piecewise-polynomial basis of Haar fast transform on Andrews were calculated on the VsiualDSP++ platform by using single-core ADSP-BF533 and ADSP-BF561 dual-core processors belonging to the Blackfin processor family using the C++ programming language. In performing the calculations, processors of the same frequency were selected, i.e., the operating frequency of the ADSP-BF533 processor core was 600 MHz and the operating frequency of each core of the ADSP-BF561 processor was 600 Mhz. A study was conducted on the values of the analytical function $y = e^x$ in order to estimate the time spent on the calculation of Haar fast transform in fractional-polynomial bases in these processors. The results of the study are presented in Table 1.

Table 1. *Comparison of computational times in single-core and dual-core processors on piecewise-polynomial basis of Haar fast transforms*

N	Piecewise constant			Piecewise linear			Piecewise quadratic		
	ADSP-BF533 (sec.)	ADSP-BF561 (sec.)	Acceleration coefficient	ADSP-BF533 (sec.)	ADSP-BF561 (sec.)	Acceleration coefficient	ADSP-BF533 (sec.)	ADSP-BF561 (sec.)	Acceleration coefficient
32	0.000102	0.000053	1.924528	0.000114	0.000059	1.932203	0.000128	0.000066	1.939394
64	0.000203	0.000104	1.951923	0.000228	0.000116	1.965517	0.000255	0.000131	1.946565
128	0.000405	0.000204	1.985294	0.000454	0.000229	1.982533	0.000508	0.000259	1.96139
256	0.000808	0.000415	1.946988	0.000907	0.000465	1.950538	0.001016	0.000526	1.931559
512	0.001614	0.000837	1.928315	0.001813	0.000939	1.930777	0.002031	0.001059	1.917847
1024	0.003226	0.001682	1.917955	0.003625	0.001886	1.922057	0.004057	0.002125	1.909176
2048	0.006449	0.003374	1.911381	0.007247	0.003781	1.916689	0.008109	0.004257	1.904863

In this table:

N is the number of array elements;

Acceleration coefficient $= \frac{\text{Single-core processor computing time}}{\text{Dual-core processor computing time}}$

VI. CONCLUSION

As can be seen from Table 1, a dual-core processor consumes 1.9 to 1.98 times less time than a single-core processor. Hence, the parallelization of calculations using multi-core processors allows to reduce the time spent and increase the efficiency compared to single-core computing.

The main advantage of developing parallel applications for dual-core processors on the VsiualDSP++ platform is that, it creates a separate subprogram for each core and shared memory. This allows different calculations to be performed in parallel at the same time in each core. The two cores can use the common data and functions represented in the L2 shared memory project in parallel at the same time.

REFERENCES

[1] Allen Dj. Architecture processors for digital signal processing. M .: Technosphere, 2006, 279p.

[2] Sotnikov A. Features of architecture and programming of dual-core processors of the Blackfin family ADSP-BF561 // Components and technologies. 2007. №6.

[3] Valpa O.D. Development of devices based on digital signal processors of the company Analog Devices with the use of Visual DSP++. - M .: Hot line - Telekom, 2007. - 270 p.

[4] VisualDSP++ 5.0 C/C++ Compiler and Library Manual for Blackfin Processors, Analog Devices, Inc. 2008.

[5] Gergel V.P. Computing performance for multi core multiprocessors system. Educational tool - Novgorod; Izd-vo NNGU i . Lobachevskiy N.I., 2010.

[6] Sato Yu. No panic! Digital signal processing: Per. s japon. - M: Dodeka - XXI, 2010, 176 p.

[7] Sotnikov A. Design with the use of processors Analog Devices. Digital KIX filter // Components and technologies. 2010. № 10.

[8] Stephen Smith. Digital signal processing.Practical guidelines for engineers and scientist. per. s angl. A. Yu. Linovicha, S. V. Vityazeva, I. S. Gusinskogo. - M .: Dodeka-XXI, 2012.

[9] ADSP-BF561 Blackfin® Processor Hardware Reference, Analog Devices, Inc. 2013.

[10] ADSP-BF533 Blackfin® Processor Hardware Reference, Analog Devices, Inc. 2013.

[11] Zaynidinov H.N.Methods and means of signal processing in piece wise -polynomial bases. Tashkent 2014, 192p.

[12] Ivanova V.G., Tyajev A.I. Digital signal processing and signal processors.educational tool . Samara: PGUTI, 2017. -252 p.

[13] https://www.analog.com/ru/products/processors-dsp/dsp/blackfin.html

[14] https://www.analog.com/ru/parametricsearch/11130#/