FPGA-based Reconfigurable Computer Systems for digital image processing

Ilya I. Levin, Igor A. Kalyaev, Alexey I. Dordopulo, Liubov M. Slasten, Vyacheslav A. Gudkov,
Vassily B. Kovalenko

Abstract— Multithread processing of large data at a rate of data receiving, which is typical of various problems of digital signal processing (locating image features by template matching, digital filtering, etc.), can be successfully performed on reconfigurable computer systems. The paper covers design principles, technical parameters and design features of reconfigurable computer systems based on field programmable gate arrays (FPGA) of Xilinx Virtex, applied for problems of digital signal processing at a rate of data receiving. A software suit, which was developed for realization of applications, written in a high-level programming language, and which provides creation of effective solutions of digital signal processing tasks on reconfigurable computer systems is considered in the paper. The distinctive features of the synthesized solutions are their high real performance, low power consumption, and practically linear growth of the real performance at increasing of hardware resource. Solutions of the problem of locating image features by template matching and the problem of processing of speckle-images using the Labeyrie method for astronomical object recognition, developed with the help of the considered software suit are given. There is also shown the real performance that was achieved for the given problem solutions.

Keywords— reconfigurable architecture, programmable soft-architecture, FPGA, reconfigurable computer system, high performance, parallel processing, pipeline processing, computer-aided circuit design, high-level programming language for reconfigurable systems, architecture description language.

I. Introduction

CONTINUOUS search of new approaches to supercomputer performance increasing is not limited by only technological achievements in development of basic components of supercomputer systems. It determines new architecture solutions such as application of field programmable gate arrays (FPGA) for solution of computational problems [1]. Vendors produce both stand-alone accelerators with one or two FPGAs and computational complexes. Such vendors as Nallatech [2] and Pico Computing [3] produce a number of accelerators and carrier boards with few (less than 4) FPGAs, which are used for design of servers and heterogeneous cluster systems by HP and IBM. The companies Convey [4] and Maxeler Technologies [5] design hybrid supercomputers on the base of their own heterogeneous cluster nodes that can contain 1-4 FPGA chips and several general-purpose processors. The company SRC [6] uses a similar solution which produces nodes, also called MAP processors for 1U, 2U and 4U racks (MAPstation). The MAPstation 1U contains one MAP processor. The MAPstation 2U contains up to three MAP processors. The MAPstation 4U can contain up to 10 various modules such as a MAP processor, a module with a general-purpose microperocessor, or a memory module.

In contrast to the abovementioned companies, the scientific team of SRI MCS SFU design supercomputers which contain printed circuit boards, united into a single computational resource. Each printed circuit board contains a set of FPGA chips [1]. The principal computing element of such RCSs is hardware resource of FPGAs, united into a single computational field by high-speed data transfer channels (LVDS and Rocket GTX). FPGAs of large-scale integration made by Xilinx (Virtex families, starting from Virtex-II). Owing to use of a single hardware resource of multichip RCS we can solve computationally intensive tasks from various problem domains, including tasks of digital signal processing, with high real performance at a rate of data receipt.

II. Hardware for problems of digital signal processing

Depending on the computational complexity of the problem we can use both reconfigurable accelerators for personal computers which contain about 10 interconnected FPGAs, and computer systems which consist of hundreds and thousands of FPGAs and which are placed in several computational racks or in a computational hall.

One of the first accelerators for personal computers (RAPC) was a RAPC-50 “Phecda” created in 2009 (see Fig. 1). It is applied for increasing of computational power of

This work was supported in part by the Russian Ministry of Education under Grant RFMEFI57814X0006.

A. Scientific Research Institute of Multiprocessor Computer Systems at Southern Federal University, Taganrog, Russia (phone: 8634-31-9092; fax: 8634-360-376; e-mail: levin@mvs.tsure.ru).

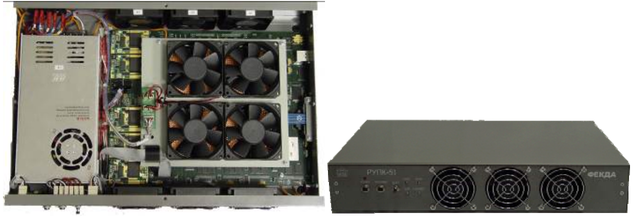
B. Scientific Research Institute of Multiprocessor Computer Systems at Southern Federal University, Taganrog, Russia (phone: 8634-36-0357; fax: 8634-360-376; e-mail: kaliaev@mvs.tsure.ru).

C. Southern Scientific Centre of Russian Academy of Sciences, Rostov-on-Don, Russia (e-mail: scorpio@mvs.tsure.ru)

D. Southern Scientific Centre of Russian Academy of Sciences, Rostov-on-Don, Russia (e-mail: lmslasten@yandex.ru).

E. Scientific Research Institute of Multiprocessor Computer Systems at Southern Federal University, Taganrog, Russia (phone: 8634-31-5491; e-mail: Slava_Gudkov@mail.ru).

personal computers for implementation of computationally intensive problems from such domains as mathematical physics, modelling and computing experiment, digital signal processing, linear algebra, etc.



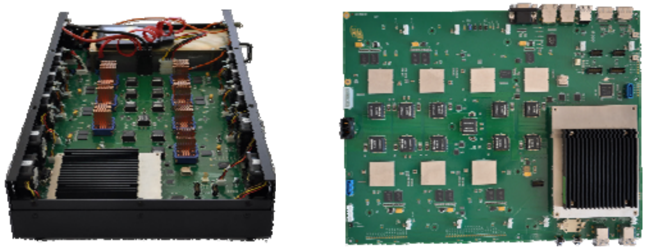
a) b)

(Fig. 1. RAPC-50)

RAPC-50 contain 16 Virtex-5 FPGAs (11 million equivalent gates each), interconnected as a lattice. Each FPGA was connected to a dynamic memory unit. The total capacity of the memory units is 1.5 GByte. The RAPC is connected to the personal computer via the Ethernet interface with the speed of data transfer of 1 GBit/sec.

The parallel program of some computationally intensive task, which must be solved on the RAPC, is prepared and debugged on the personal computer. The files of initial data for the parallel program are generated on the personal computer also. The executable file of the task and the initial data are loaded from the PC into the RAPC via the Ethernet interface, and the results are transferred into the PC. The peak performance of the RAPC-50 is 50 GFlops, and its real performance during execution of tasks of digital signal processing (processing of speckle-images using the Labeyrie method for astronomical object recognition) described in 5.1 is 35 GFlops.

The next step of development of the concept of reconfigurable accelerator design was creation of a reconfigurable computer “Celaeno” (see Fig. 2) which is applied for independent functioning with processing of data which is transferred via the Gigabit Ethernet data channel without support of IP-protocols.



a) b)

(Fig.2)

In contrast to the RAPC-50, Celaeno contains not only 6 Xilinx Kintex XC7K160T FPGAs (16 million equivalent gates) but also a control processor unit of the family COM-Express by Kontron, set on the printed circuit board. Owing to this Celaeno can be used without connection to the host-computer. All FPGAs of the personal reconfigurable computer (PRC) are interconnected as a lattice by means of LVDS channels. All FPGAs of the PRC are connected to the dynamic memory units of 256 MByte each.

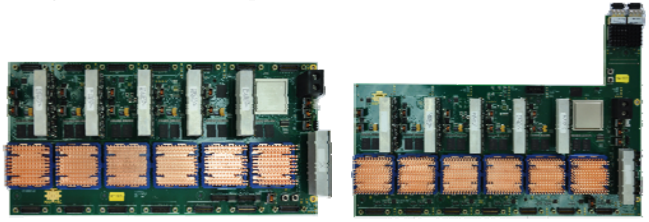
The peak performance of the reconfigurable computer block Celaeno during execution of operations on 32-digit floating point data at frequency of 330 MHz is 150 GFlops, and 75 GFlops for operations on 64-digit floating point data.

For computationally intensive tasks of large and super large dimension it is possible to use computer systems which contain hardware resource of all FPGAs united into a single computational field by high speed data transfer channels. The example of such system is a reconfigurable computer system RCS-7 on the base of Virtex-7 FPGAs. It contains a computational field of 576 Virtex-7 XC7V585T-FFG1761 FPGAs (58 million equivalent gates each), united within a 47U computational rack with the peak performance 10^{15} fixed-point operations per second.

The principal structural component of the RCS-7 is a computational module 24V7-750 (Pleiad), which can be placed into a standard 19" computational rack, and which consists of:

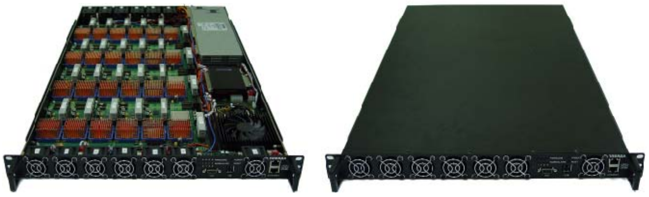
- 4 printed circuit boards of the computational module 6V7-180 (see Fig. 3);
- a control module CM-7;
- a power supply subsystem;
- a cooling subsystem, etc.

Fig. 4 shows the computational module 24V7-750.



a) b)

(Fig. 3)



a) b)

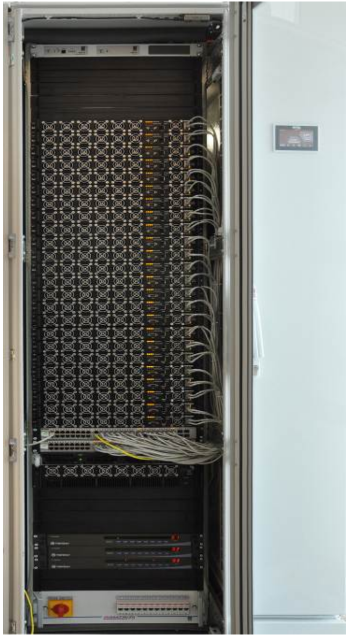
(Fig. 4)

The board of the computational module 6V7-180 contains:

- a computational field, which consists of 6 Xilinx Virtex-7 XC7V585T-1FFG1761 FPGAs, connected sequentially by 144 LVDS differential lanes running at frequency of 800 MHz;
- a controller of the board of the computational module, based on Xilinx XC6V130T-1FFG1156C FPGAs;
- 12 LVDS channels running at frequency of 800 MHz, 25 differential pairs each (SS4 connectors), applied for connection with other computational modules;
- blocks of general and reserve FPGA programming through the interfaces JTAG-1 and JTAG-2;
- a synchronization subsystem (generators ECS-2033-250-BN and clock buffers IDT5T9316NLI);

- a distributed memory which consists of 12 DRAM chips (MT47H128M16HR-25E, 128 M*16 and with the read/write frequency up to 400 MHz). Each FPGA of the computational field and of the controller of the computational module's board are connected to two DDR2 memory chips. The RAM size of the computational module's board is 3 Gbyte;
- 2 LVDS channels (20 differential pairs each) for connection with a personal computer and external devices;
- a subsystem of FPGA loading;
- a power supply subsystem, which includes DC-DC voltage converters, which provide the supply voltages: +1 V for FPGA cores; +2.5 V for the clock generator; +1.8 V for DDR2 memory chips, +3.3 V for FPGA buffer stages.

The performance of the one board 6V7-180 is 645.9 GFlops for 32-digit floating point data, and the performance of the computational module 24V7-750 is 2.58 TFlops for 32-digit floating point data. The performance of the RCS-7 (see Fig. 5), when it contains 24-36 computational modules 24V7-750 is 62-93 TFlops for 32-digit floating point data and 19.4–29.4 TFlops for 64-digit floating point data.



(Fig.5)

The application area of the RCS-7 and computer complexes designed on its base is digital signal processing and multichannel digital filtration.

III. RCS-7 SOFTWARE

The majority of existing commercial CAD-systems such as Xilinx ISE, Altium Designer, etc. provide work with only one FPGA chip within one project. Therefore, if it is necessary to design a configuration for several interconnected FPGA chips, the circuit engineer has to distribute all components of the computing structure, which corresponds to the algorithm of the solving task, between different projects, which will correspond to certain FPGA chips of the multichip RCS. Besides, he has to take into account the RCS topology and connections between FPGA chips. A large number of features of RCS architecture, topology and components, which are to be taken into account by the circuit engineer, significantly complicates design of RCS configurations, required for the task which is solved on the multichip RCS, and practically rules out porting of the complete solution to some other RCS which has a different architecture or configuration. The RCS programming consists of two stages. At the first stage, the circuit engineer creates the computing structure for the task. At the second stage, the application programmer develops the parallel program, which determines data flows in the designed computing structure. That is why the time period, usually needed for creation of the RCS task solution, is rather long (4-9 months).

To program RCSs, designed and implemented in Scientific Research Institute of multiprocessor computer systems at Southern Federal University (Taganrog, Russia), we use a software suit, developed by our team of scientists [1, 7]. The software suit is based on structural-procedural methods of organization of computations and determines both the structure of the computer system within the field of FPGA logical cells, and organization of parallel processes and data flows. Principles of RCS programming are based on the principles of structural procedural organization of computations and are the same for all generations of RCSs designed in SRI MCS SFU. So, applications for the RCS-7 are developed with the help of the high-level programming language COLAMO [1, 7]. After translation of the program the configuration of the computer systems is created automatically, and it consists of bitstream files of all FPGAs (the structural component of the parallel application). The other part of translation result is the parallel program, which controls data flows and organization of computations in the RCS. The distinctive features of the software suit based on the programming language COLAMO in comparison with the development tools MitrionC [8] and CatapultC [9], are automatic mapping, synchronization and generation of configuration for multichip RCSs, 60-90% of used hardware resource of each FPGA chip and high frequencies (250 – 350 MHz).

The basic components of the software tools of application development with the help of the high-level programming language COLAMO:

- translator of the programming language COLAMO, which translates the source COLAMO-program into the information graph of the parallel application;
- synthesizer of multichip circuit solutions FireConstructor, which maps the information graph, generated by the translator of the programming language COLAMO, on the RCS architecture, places the mapped solution into the FPGA chips and automatically synchronizes all fragments of the information graph in different FPGA chips;
- library of IP-cores, which correspond to the COLAMO instructions and are all-in-one structurally implemented circuit solutions for various problem domains,

and interfaces, applied for data processing speed matching and for connection into the single computing structure.

The unique feature of the RCS-7 software suit is support of special-purpose soft-architectures, which are applied for creation and programming of macroobjects. A macroobject is a set of computational nodes which perform a certain set of operations, and which are united by some communication system. For each class of tasks, solved in RCS, we can select a certain set of optimal computing structures (macroobjects), which provide the most effective solution of the tasks of the given class. It is possible to change the number of functional nodes and parameters of any functional node (such as the operand capacity, the number of data channels, the instruction set, etc.) of any macroobject, but their functions cannot be changed. So, from the viewpoint of the RCS programmer the macroobject is a “pattern” which can be redefined by the RCS programmer during design of some engineering solution, and then it can be replicated in FPGAs of computational modules in the required quantity and be connected with similar or any other macroobjects into computing structures, which optimally correspond to the structure of the solving task.

On the base of macroobjects we can create various RCS soft-architectures. An RCS soft-architecture is a computing structure, designed by the circuit engineer. It consists of macroobjects and allows modification of connections between its components only by program re-adjustment and without re-programming of the FPGA chips of the computational field (no re-loading of FPGA bitstream files). So, we can create any computing structures, which are required for task solving.

Owing to soft-architectures, fundamental programming principles and high-level RCS programming language it is possible to adapt very easily all software RCS development tools to any new RCS architectures and topologies of computational modules with no cardinal modifications of the source code of any component of the software suit and to reduce the time of task solution.

IV. SOFT-ARCHITECTURE PROGRAMMING FOR PROBLEM DOMAINS

For creation of soft-architectures we have developed a language SADL (Soft-Architecture Development Language)[7]. The SADL-description of the soft-architecture is translated into a virtual architecture of the computer system. Then the synthesizer FireConstructor maps the information graph of the task on the virtual architecture.

Creation of any soft-architecture requires the following actions:

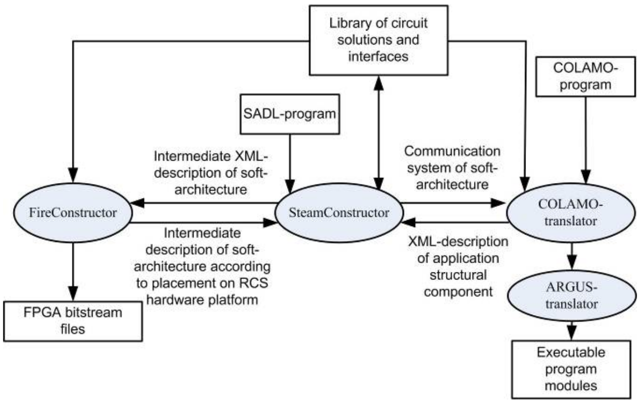
- creation of a SADL-description of the soft-architecture;
- translation of the soft-architecture description into an intermediate description with the help of the synthesizer FireConstructor;
- mapping of all soft-architecture components on the hardware platform with the help of the synthesizer of scalable parallel-pipeline procedures SteamConstructor.

The SADL-translator converts the program into an intermediate form, which is placed by the synthesizer FireConstructor on the RCS hardware platform. The result of placement is a modified file of the intermediate form and bitstream files of all FPGA chips, occupied during placement of the soft-architecture on the RCS hardware platform. After successful placement of the soft-architecture on the RCS hardware platform, it can be used for solving of various tasks of the certain problem domain.

Development of any parallel application on the base of the RCS soft-architecture requires the following actions:

- to develop a parallel application in the high-level language COLAMO;
- to translate the parallel application and to generate structural and procedural components;
- to map the structural component of the parallel application on the soft-architecture using the synthesizer SteamConstructor;
- to translate the procedural component of the parallel application into assembler instructions of soft-architecture components;
- to generate executable file of the parallel application (with instructions for the soft-architecture components) for loading into the RCS;
- to load FPGA bitstream files, which were generated during placement of the soft-architecture components on the RCS hardware platform, into the RCS;
- to load the executable file of the parallel application;
- to load input data of the task into the RCS soft-architecture;
- to run the application and to save calculation results.

Figure 6 shows interaction of application development software tools.


Library of circuit solutions and interfaces
COLAMO-program
SADL-program
Intermediate XML-description of soft-architecture
Communication system of soft-architecture
FireConstructor
SteamConstructor
COLAMO-translator
Intermediate description of soft-architecture according to placement on RCS hardware platform
XML-description of application structural component
ARGUS-translator
FPGA bitstream files
Executable program modules

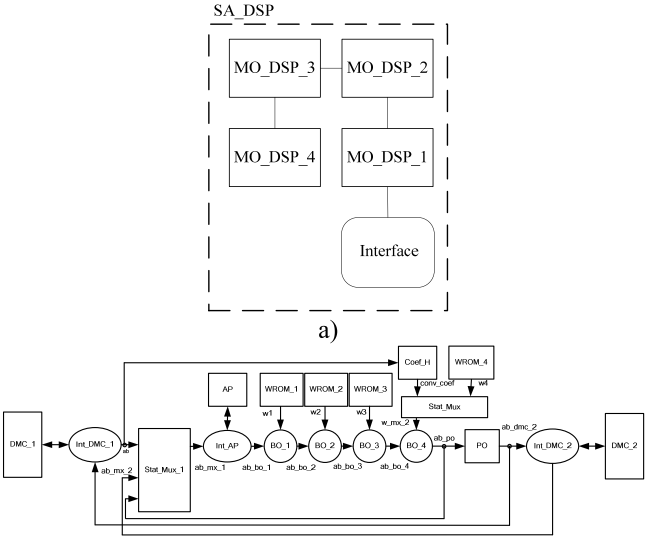
(Fig.6)

Owing to the created software tools, development and modification of soft-architectures requires no highly qualified circuit engineer for modification of the computing structure. The time, which is usually needed for design or modification of the soft-architecture is significantly reduced, and the effectiveness of the generated multichip architectural solutions is comparable with solutions, implemented manually by circuit engineers. According to the basic principles of the language COLAMO parallel applications can be easily modified and adapted to the available computational resource.

Owing to automatic mapping of information graphs on the RCS hardware resource, the application programmers can consider the RCS as a virtual FPGA with a huge number of logic cells, but not as a set of FPGA chips. Using the developed software suit, the RCS programmer can develop and debug RCS parallel applications, paying no attention to the features of the RCS architecture.

V. Solving Digital Signal Processing Problems with the Help of Soft-Architectures

For implementation of digital signal processing (DSP) algorithms we have developed a DSP soft-architecture based on MO_DSP macroobjects (see Fig. 7a) which can be used for solution of various tasks from the given problem domain and does not require reloading of FPGA configuration bitstream files.



a)

b)

(Fig.7)

Each MO_DSP macroobject (see Fig.7b) is a standalone computational unit which can be adjusted independently during implementation of the following typical DSP algorithms: fast Fourier transform, signal filtering by the method of numerical sequence convolution, algorithms of image processing [10]. Each MO_DSP macroobject consists of:

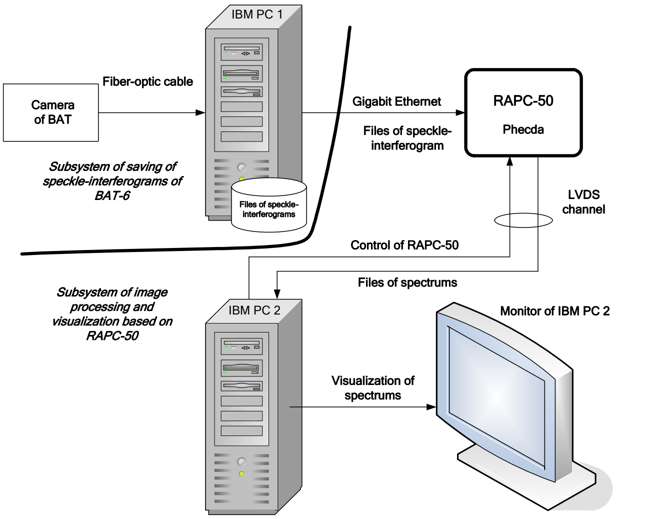
- distributed memory controllers (DMC_1, DMC_2) which yield control signals and data streams;
- a DMC interface (Int_DMC_1);
- an address processor (AP) which generates a continuous stream of addresses for RAM read/write operations [11];
- an address processor interface (Int_AP);
- a FIFO memory within the nodes BO_i (FIFO_1);
- a memory for convolution coefficients (Coef_H);
- adders/subtractors (ALU_1, ALU_2);
- multipliers (FU_1, FU_2) [12];
- static multiplexers (Stat_Mux, Stat_Mux_1, K);
- a functional scaling unit (FU_M);
- a dual-port memory for accumulation of results (Mem_PO).

The computational structure of the macroobject is a pipeline whose principal stages are BO_i nodes which perform data processing according to digital signal processing algorithms. Each node can be adjusted to any iteration of fast Fourier transform (FFT), reverse fast Fourier transform (RFFT) or to “transit” operation. Besides, the memory WROM_4 connected to the node BO_4 provides multiplication by convolution coefficients or by impulse response of the implemented filter.

The macroobject also contains the address processor AP owing to which it is possible to perform high dimensional FFT in several cadrs because readjustment of the pipeline to new iterations takes not more than 100 cycles. The address processor is typical for computing structures within which DSP tasks are implemented.

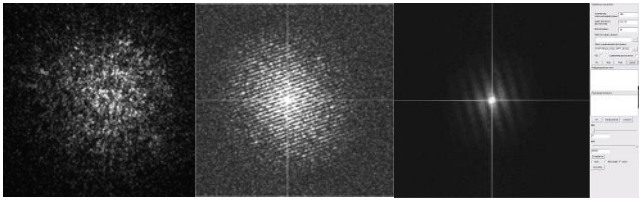
5.1. Solving the problem of speckle image processing using the Labeyrie method of astronomical object identification

According to the image processing algorithm based on the Labeyrie method, lines and rows of the image are processed sequentially. The calculated spectrum is saved together with spectrums of the images captured on the previous iterations of the algorithm. Images of astronomical objects are made by the camera which is connected to the biggest telescope of Russia and Eurasia – optical “Big Altazimuth Telescope” (BAT). Then the images enter the reconfigurable computer system with frequency 20-28 Hz depending on the shutter speed of the camera of the telescope. The accumulated results are extracted and visualized after each n images. Using the RCS for observation of multiple stars the scientist can receive information about the observable object in real-time and timely make decision whether to continue observations or not. Fig. 8 shows the structure chart of the speckle interferometry system.


IBM PC 1
Camera of BAT
Fiber-optic cable
Subsystem of saving of speckle-interferograms of BAT-6
Files of speckle-interferograms
Gigabit Ethernet
Files of speckle-interferogram
RAPC-50
Phecda
LVDS channel
Control of RAPC-50
Files of spectrums
Subsystem of image processing and visualization based on RAPC-50
IBM PC 2
Monitor of IBM PC 2
Visualization of spectrums

(Fig.8.)

As a hardware platform for the problems solved with the help of the DSP soft-architecture we used the reconfigurable accelerator for personal computers RAPC-50 “Phecda” described in the unit 2. The system of observation of multiple star systems contains two personal computers IBM PC1 and IBM PC2, the camera of the BAT telescope, RAPC-50. The camera of the telescope captures the image and transfers it via optical channels to the IBM PC1. The computer IBM PC1 stores the received images and transfers them via Gigabit Ethernet to the accelerator RAPC-50, which processes the images according to the required algorithm. When the current image has been processed, RAPC-50 transfers the result via LVDS-channels to the personal computer IBM PC2 for its visualization and estimation of the speckle-interferogram.



a) b) c)

(Fig.9.)

The application of visualization contains the following graphic options: zoom of the captured image, calculation of the brightness of the points along the line set by the user, the brightness in the selected point of the spectrum, variations of upper and lower limits of imaging brightness, saving of the accumulated spectrum in the JPEG-format. All mentioned tools help the user to analyse the accumulated spectrum in details and get complete information from it.

The real performance, achieved during processing of speckle-images according to the Labeyrie method on the RAPC-50 is 35 GFlops. Owing to use of MO_DSP soft-architecture it was possible to reduce the time of algorithm development in 2.1 times in comparison with its development using VHDL.

5.2. Solving the problem of locating image features by template matching

We have solved the task of locating image features on RCS-7 with the help of soft-architectures. The solution of this task is based on the method of template matching. The template is moving along the image from left to right and from top to bottom and it is being compared with all features of the image. As an evaluation criterion we use cross correlation between the initial image and the template. The position of the correlation maximum corresponds to the position of the located feature. As the views of the target feature may be considerably different, it is necessary to define not only its position, but also to measure similarity between the target feature and each template, and only then the final decision can be made. The input image is partitioned into sections which are processed one by one with overlapping. If it is necessary, zero bits will be included into the last section to make its length proper.

During implementation of the task of locating image features on the RCS-7 on the base of the method of template matching, we used the following basic parameters:

- the size of the input image is 512×512 (or 544×544, if we take into account additional zeroes);
- the size of the used templates is 32×32;
- the size of the selected section is 64 (32 zeroes are added to the last section);
- the number of sections within the image is 256;
- the size of overlapping between sections is 32.

The task was implemented on the RCS-7, which contained 24 24V7-750 computational modules. The real performance which was reached during execution of the task was 41.8 TFlops or 67.4% from the peak performance.

VI. Conclusion

RCSs are advanced direction of development of high-performance computer systems, owing to which the user can create virtual special-purpose calculators within the basic architecture. In addition, the structure of such special-purpose calculator is similar to the structure of the solving task. Application of soft-architectures for solving applied problems provides reduction of the time of development and debugging of the algorithm. Besides, it also provides high effectiveness of calculations and practically linear growth of performance of increasing computational resource.

References

[1] I.A. Kalyaev, I.I. Levin, E.A. Semernikov, V.I. Shmoilov. (2012). Reconfigurable multipipeline computing structures. Nova Science Publishers, New York.

[2] http://www.nallatech.com/

[3] http://picocomputing.com/

[4] http://www.conveycomputer.com

[5] http://www.maxeler.com/

[6] http://www.srccomp.com/

[7] V.A. Gudkov, A.A. Gulenok, V.B. Kovalenko, L.M. Slasten. Multi-level Programming of FPGA-based Computer Systems with Reconfigurable Macroobject Architecture // *IFAC Proceedings Volumes (ISSN 14746670), Programmable Devices and Embedded Systems*, Volume №12, part №1, 2013, pp. 204-209

[8] http://www.mitrionics.com

[9] http://calypto.com/en/products/catapult/overview/

[10] Ali M. Reza Adaptive Noise Filtering of Image Sequences in Real Time // WSEAS Transactions on Systems, Print ISSN: 1109-2777, E-ISSN: 2224-2678, Volume 12, 2013, Issue 4

[11] J. L. Mazher Iqbal, S. Varadarajan. New Approach to Memory Less Design and Look-Up-Table Realization for Low-Complexity Reconfigurable Digital FIR Filter Architectures // WSEAS Transactions on Systems, Print ISSN: 1109-2777, E-ISSN: 2224-2678, Volume 12, 2013, Issue 3

[12] Wai-Leong Pang, Kah-Yoong Chan, Sew-Kin Wong, Choon-Siang Tan VHDL Modeling of Booth Radix-4 Floating Point Multiplier for VLSI Designer's Library // WSEAS Transactions on Systems, Print ISSN: 1109-2777, E-ISSN: 2224-2678, Volume 12, 2013, Issue 11.