Image and data processing using reconfigurable computer systems

ALEXEY DORDOPULO[1*], ANDREJ GULENOK[2**], VIACHESLAV GUDKOV[3**], IGOR KALYAEV[4**], ILYA LEVIN[5**]
[*]Southern Scientific Centre of the Russian Academy of Sciences
41 Chekhov Ave., Rostov-on-Don, 344006
[**]Kalyaev Scientific Research Institute of Multiprocessor Computer Systems at Southern Federal University
2 Chekhov St., Taganrog, Rostov region, 347928
RUSSIA
[1] Chesshire@yandex.ru, [2] andrei_gulenok@mail.ru, [3] slava_gudkov@mail.ru, [4] niimvs@yandex.ru, [5] niimvs@yandex.ru

Abstract: - Multithread processing of large data at a rate of data receiving, which is typical for various problems of digital signal processing, can be successfully performed on reconfigurable computer systems. The paper covers design principles and features of reconfigurable computer systems based on field programmable gate arrays (FPGA) of Xilinx Virtex, applied for astronomical object identification and comparative analysis of existing tools for programming the single-chip and multichip reconfigurable computer systems. A software suit based on high-level programming language COLAMO which provides creation of effective solutions for speckle-images processing using the Labeyrie method for astronomical object recognition 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.

Key-Words: - reconfigurable architecture, FPGA, reconfigurable computer system, high performance, parallel processing, pipeline processing, computer-aided circuit design, high-level programming language for reconfigurable systems

1 Introduction

In general, problems deal with collection, real-time processing and analysis of huge data streams according to fixed algorithms may be solved by means of various methods. One of such methods is design of a special-purpose computer system for problem implementation where each operation of the problem (or let us call it task) is carried out by specially assigned computing element. All elements of the computer are connected according to connections between vertices of the task graph. Such method has obvious advantages. The solution will be obtained during minimum time with maximum specific performance. The ratios are practically equal to unity. However, disadvantage of the method is complicated, laborious and expensive design and manufacturing of the special-purpose computer.

Another solution is implementation of the problem on a general-purpose multiprocessor computer system with fixed architecture (a cluster system). In this case each operation of the task graph is assigned to a processor of the system. Connections between processors are created, taking into account connections between vertices of the task graph and restrictions of the system fixed architecture. The main advantage of the method is relatively low price of the cluster system. But there are several considerable disadvantages such as high non-productive time which is needed to organize computational process, but not to perform calculations. Besides, processing time of the task and the number of used processors grow and specific performance drops down.

The data flow problem may be implemented by means of a reconfigurable multiprocessor computer system. In this case the task is mapped on a computational field of computing elements by means of tools of structural or so-called circuit programming. Processing time is practically equal to that of special-purpose computer system. Specific performance is high and it grows when size of computational field increases. The only

disadvantage is relative complexity of structural programming of the computational field.

Reconfigurable computer systems (RCS) are widely used for solving of computationally laborious tasks of various domains of science and technology. In comparison with cluster multiprocessor computer systems RCSs have several significant advantages such as high real and specific performances during execution of tasks, high energy effectiveness, etc. Vendors produce both stand-alone accelerators with one or two FPGAs and computational complexes. Such vendors as Nallatech [15] and Pico Computing [16] 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 [5] and Maxeler Technologies [13] 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 [21] 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. The most reasonable method of reconfigurable computer system design is use a set of FPGA chips as a principal computational element. The FPGA set is united into a computational field by high-speed data transfer channels. Owing to such union we can get a powerful computing resource, sufficient for effective solving of the task.

2 Single-chip RCS programming tools

At present many various design systems allow design structural solutions of applied tasks on FPGAs. The most popular software systems given by FPGA vendors are: ISE and Vivado by Xilinx [21], Quartus II by Altera [17] and Actel Libero IDE by Actel [12]. Besides the design environment of digital devices, these tools contain several utility tools such as timing analyzers, placing editors, FPGA programming units, simulators of digital devices, etc. Owing to various tools these design systems provide a complete cycle of digital device design within a single FPGA chip: creation of initial project description, synthesis, simulation, placing, tracing and configuring of the chip.

Design tools for digital devices in FPGA chips are developed by outside vendors. These vendors are not engaged directly the production of FPGA chips. So, the integrated computer-aided design system of radioelectronic devices Altium Designer [4] was developed by the Australian company Altium. Besides the designed tools of printed circuit boards, this system includes tools for designing of the electronic devices based on FPGAs. In this case the methodology of the project is proposed for the developer, similar to the development of printed circuit boards, and independence from FPGAs vendors.

Disadvantages of these systems are lack of support for development of multichip solutions, as well as a long development time.

In the early 2000s, the company Starbridge Systems had developed reconfigurable computers called Hypercomputer Systems based on Xilinx FPGAs. For programming these reconfigurable computers is used a unique multichip graphic design tool – Viva.

The distinctive features of the Viva system were support of multichip solutions, recursive resource scan, typed data at the inputs of computing and the control blocks. These features differed Viva system from usual graphic design tools (Xilinx ISE, Altera Quartus II, Actel Libero) [2]. The implemented synchronization system for data flows and control signals are often used significant additional hardware resource. Developed in Viva system RTL synthesizer was lost in 1.5-3 times comparing with Xilinx ISE on optimization of an existing resource [20]. However, in spite of several contracts with NASA, Starbridge Systems ceased to exist and support of Viva system.

Due to continuous growth of FPGA capacity design of the applied task solutions in FPGA chips with the help of hardware description languages (VHDL, AHDL, Verilog, etc.), and design of digital devices in graphic editors becomes more and more complicated. That is why at present the leading vendors of FPGAs and reconfigurable computers orient to high level languages. So, the new development environment Vivado by Xilinx contains a new design tool Vivado HLS [4], based on a high level language, and Altera suggests Altera SDK [2] for its FPGA for a new standard of parallel programming of OpenCL heterogeneous systems. These solutions use translators of C-like languages which generate code in hardware description languages on the level of register transfers from a program written in a C-like high level language (RTL-level) (C-to-RTL translators).

In spite of syntax similarity of these C-like languages with the language C such approach does not mean that the initial program written in C for a

personal computer or a cluster computer system will be understood by C-to-RTL translators. The language C was chosen as a base because it is a wide spread language, and this fact considerably simplifies mastering of new development tools of application development for FPGAs. Also, if we use C-to-RTL translators then the whole program or explicitly marked procedures are translated into RTL-descriptions of single FPGA chips. Such development systems have no tools which provide automatic partitioning of the parallel program into several interconnected FPGA chips.

In Vivado HLS the project is designed within a single FPGA chip, and if the developer needs more hardware resource than the FPGA chip has, he himself must distribute calculations between several projects for each FPGA and synchronize control and data flows between them.

The OpenCL standard is used by Nallatech for programming solutions on FPGA, which are used as coprocessors in hybrid computer systems. Nallatech is the vendor of the reconfigurable accelerator with one or two FPGAs by Altera. The OpenCL standard implies using of several FPGAs in one project. In this case implementation of applications in FPGAs is provided by functions which are called from the libraries of the Altera SDK, and in each FPGA involved into computational process, the calculations, described by a fragment of code, are performed. So, the application, written according to the OpenCL standard, is a basic code, written for traditional processors, and a number of fragments, written for FPGAs which are used as co-processors. In this case the developer has to synchronize data flows himself.

3 Multichip RCS for digital signal and data processing

In contrast to the abovementioned single-chip RCS vendors, 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 [15]. The principal computing element of such RCSs is hardware resource of FPGAs, united into a single computational field by high-speed data transfer channels.

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 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.

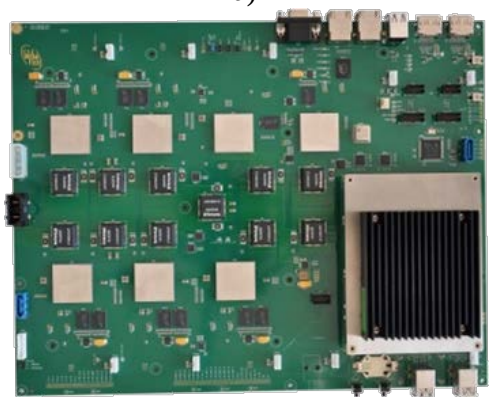
The parallel program of some computationally intensive task, which must be solved on the RAPC, is prepared and debugged on the personal computer. The files of initial data for the parallel program are generated on the personal computer also. The executable file of the task and the initial data are loaded 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 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)

ISBN: 978-1-61804-344-3

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 Reconfigurable computer “Celaeno”

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.3a);
- a control module CM-7;
- a power supply subsystem;
- a cooling subsystem, etc.

Fig.3b shows the computational module 24V7-750.

a)

b)

Fig.3 Reconfigurable computer module 24V7-750

On the base of CM Pleiad, in 2013 we had designed a reconfigurable computer system RCS-7, which contained 24 computational modules, and which can be extended up to 36 computational modules. The performance of RCS-7, when it contains from 24 to 36 24V7-750 CMs is from 62 to 93 TFlops for processing of 32-digit floating point data, and 19.4÷29.4 TFlops for processing of 64-digit floating point data, respectively. Fields of application of RCS-7 and RCS-7-based computer complexes are digital signal processing and multichannel digital filtering [3, 14].

ISBN: 978-1-61804-344-3

4 Language COLAMO and multichip RCS programming suit

An RCS programming suit in the high-level language COLAMO [9] is designed in SRI MCS SFU. The distinctive feature of this complex is the support for multichip FPGA programming in a single project. Thus, it doesn't specify explicitly in the program which parts of the code will be executed in any FPGAs. A synthesizer Fire!Constructor is a part of the developed programming suit. During compilation it divides automatically the structure of the parallel program into disjoint fragments and automatically maps them on the multichip RCS hardware resource.

The synthesizer Fire!Constructor works with representation of the parallel program as information graph. The input data for the synthesizer Fire!Constructor are the library of circuit elements, the structure of the multichip RCS and the description of the information graph of the parallel program.

The library of circuit elements is a set of elements, developed in the languages VHDL and Verilog or in existing graphic environments, for specific FPGA families. These elements perform various arithmetic and logic operations, control of information flows, provide functions of the distributed RCS memory controllers, etc. All elements of the library have description. The components indicated in this description are: the supposed hardware resource, that occupied by any element in the FPGA, the maximum clock frequency, calculated by element, the purpose and capacity of the element outputs, latency (for conveyor blocks) or delay (for procedural blocks), the purpose and capacity of outputs. An example of description of a library element is given in the Fig. 4.

```
--CFC Version 2.0
--CFC ElementFPGA V7
--CFC ElementName sum_fxs_32x32_rg
--CFC ElementTypeProcessing pipeline
--CFC ElementFrequency 400
--CFC ElementResource : Lt, Dl, FF, LUT, LUTm, BRAM, DSP
--CFC Resource : 3,0,67,78,22,0,0
entity sum_fxs_32x32_rg is
  port ( clk : in  std_logic;                    --CFC -CLK_In
       X   : in  std_logic_vector (31 downto 0); --CFC -Data_IN -0 -
fxs
       Y   : in  std_logic_vector (31 downto 0); --CFC -Data_IN -1 -
fxs
       Z   : out std_logic_vector (31 downto 0)  --CFC -Data_OUT -
0 -fxs
       );
end sum_fxs_32x32_rg;
```

Fig. 4 Description of an integer signed 32-bit adder

The example corresponds to the description of an integer signed 32-bit adder. The type of the FPGA family "V7" (Xilinx Virtex-7), designed for this circuit block, and the element name are given in this figure. The parameter "pipeline" shows that this block is a pipeline; the guaranteed maximum operating frequency of element is 400 MHz. The latency of the block is equal to 3. The expected hardware resource is 67 triggers, 78 logical blocks (conversion LUT tables), 0 blocks of distributed memory, and 0 blocks of signal processors DPS. The output "clk" is indicated as the input of clock frequency. The outputs "X" and "Y" are indicated as inputs of operands, "Z" as an output result.

From the programmer's point of view the RCS structure contains the description of FPGA chips (hardware resource, names of various service FPGA outputs – a discharge, clock frequency, indication, etc.), included in the RCS, and the description of user connections between the FPGA chips (the capacity of the connections and the FPGA output names). The computational module "SKAT-7" and its structure are given in the Fig.5.

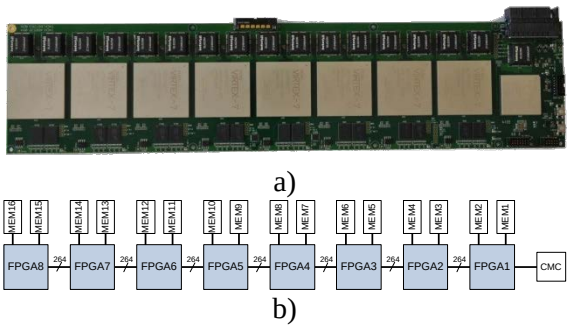


Fig.5 Appearance of the computational module "SKAT-7" (a); its structure (b)

This computational module contains 8 high-performance FPGA Xilinx Virtex-7 XC7VX585T chips (48.5 million equivalent gates each), 16 blocks of distributed memory of total volume 4 GB and the controller of computational module that implemented the interface of data and control signal exchange between the computational module and the control host-machine.

5 Astronomical object identification using the Labeyrie method for speckle image processing

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

ISBN: 978-1-61804-344-3

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.6 shows the structure chart of the speckle interferometry system.

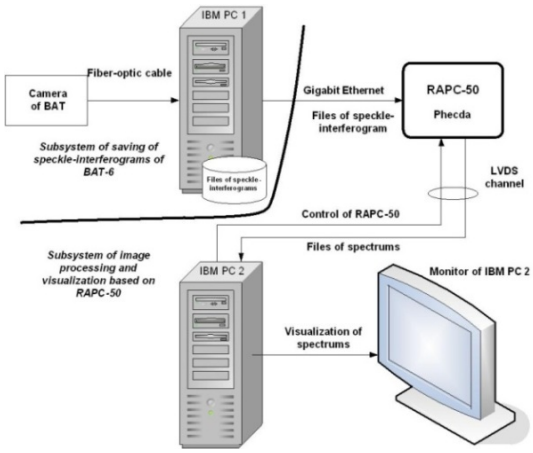
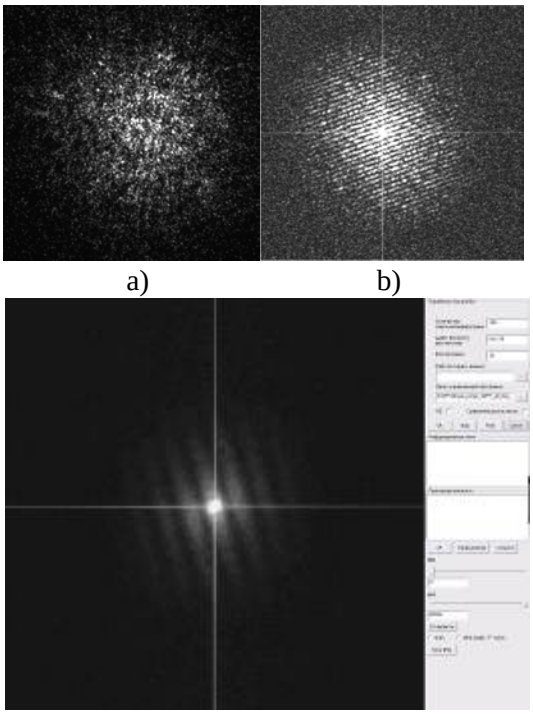


Fig.6 Structure of the speckle interferometry system

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 3. 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.

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 (see Fig.7). All mentioned tools help the user to analyse the accumulated spectrum in details and get complete information from it.



a) b)

c)

Fig.7 Series of speckle-interferogram

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 designed software suit it was possible to reduce the time of algorithm development in 2.1 times in comparison with its development using VHDL.

6 Comparative analysis of RCS programming tools

For comparative analysis of multichip RCS programming suits we use a standard symmetric encryption DES algorithm (Data encryption Standard). For this algorithm the researches of solution efficiency on FPGA were performed frequently using programming languages such as Mitrion-C, Handle-C, Impulse-C [1, 6, 18].

Hardware implementation of standard encryption DES algorithm was developed using the high-level language COLAMO. The fragment of the COLAMO-application is given in the Fig.8. The full COLAMO-text of this program contains about two thousand lines.

ISBN: 978-1-61804-344-3

```
Program DES;
Include V7_Skat.Skat, V7_Cryptography.V7_Cryptography;
…
// MAIN SUBCADR
SubCadr DES_top(In : sPT,sKey_const,sKey_var; Out : sCT);
  GetKDelay(sKey_var,kdelay[*,*]);
  for jv:=0 to NP-1 do
    DES_enc(sPT[*],   // plain text
    sKey_const[jv],kdelay[*,*],
    sCT[*]            // cipher text
    );
EndSubCadr;

Cadr Cadr1;
for iv:=0 to NC-1 do
begin
…
   for s:=0 to n-1 do
   begin
      // 36-bit var part key
      key_in[iv,s]:=DataIncrement(param[iv,0],1,n);
      DES_top(text[iv,*],      // plain text
      key_consts_com[iv,*],
      key_in[iv,s],
      ciphertext[iv,*]);
…
      res[iv,s]:= ciphertext[iv,s];
…
   end;
end;
EndCadr;
End_Program.
```

Fig.8 A fragment of the program COLAMO-text

The text of the developed program contains no explicit instructions of any calculations which must be performed in any FPGA. The library "V7_Skat.Skat" was specified in the header. The library only indicates that this program must be performed on the computational module "SKAT-7".

During translation of the parallel program the COLAMO-translator forms a stream component. The stream component contains the following commands: commands for distributed memory controllers and controllers of data exchange with the controller of the computer module, commands of data download/upload of data and control commands of the computing – start, stop, synchronization, discharge, etc. Also the structural component is formed. It's a virtual computational device of library elements which performs the applied task according to the parallel program.

The structural component is transmitted to the synthesizer Fire!Constructor, which performs automatic mapping of virtual computational devices, described as of an information graph, on disjoint fragments. These fragments are structurally implemented in FPGA chips of the target RCS. The problem of automatic mapping of the parallel program on hardware resource of the multichip RCS consists of three steps: partition of the information graph into disjoint subgraphs, placing of the subgraphs into RCS FPGAs and tracing of the external connections of the placed subgraphs according to the communication system of the RCS. Each of these subproblems is solved by several algorithms with sets of various selection criteria for optimal solutions.

As a result of placement of the virtual computational device on hardware resource of the computational module "SKAT-7" 220 devices were placed in each FPGA which implemented the encryption DES algorithm. The information graph contained more than three million vertices.

The result of the synthesizer Fire!Constructor is the VHDL-files of descriptions and the files of time and topological constraints (User Constraints Files). The VHDL-files describe the structural implementations of the fragments of the parallel program. These files and the library of circuit elements are the base for the ISE synthesizer which create projects and then generate bitstream files for each FPGA. Then the bitstream files are loaded into the RCS.

The control programs provide loading of the initial data, start of the application and uploading of the results, which are written separately for each parallel program in C++ and C# languages. These programs are executed on the control computer, connected to the RCS. The COLAMO-translator automatically generates control programs for translation of application. As a rule, the numbers of these control programs is sufficient to manage the calculations in the RCSs. However, an experienced user can modify the control program which implements additional control RCS calculations.

The characteristics of hardware implementation of the standard encryption DES algorithm which were performed using multichip RCS programming suit are given in the Table 1.

Name	Value
Number of vertices in graph	3 103 456
Filling FPGA by triggers	94%
Filling FPGA by logic	92%
Working clock frequency	250
Number of devices	1 760

Table 1 The characteristics of solutions of the model task

The results of comparative analysis of the system software on the basis of the Mitrion-C, Handle-C, Impulse-C and COLAMO-Fire!Constructor are given in the Table 2.

ISBN: 978-1-61804-344-3

Feature	Mitrion-C	Handle-C	Impulse-C	COLAMO
Programming in high-level language	Yes			Yes
Complexity of programming	Low			High
Support of multichip programming	No			Yes
Support of conveyor calculations	Yes			Yes
Frequency of synthesized solution, MHz	100	120	200	250
Number of logical blocks, involved by one device	1912	2047	2234	1269
Number of triggers, involved by one device	1408	1682	1888	2594
Maximum number of devices in a single FPGA XC7VX585T	119	112	102	220

Table 2 Comparison of the characteristics of the standard encryption DES algorithm RCS implementations

The more the number of triggers, used in the COLAMO program, the more the frequency of the synthesized solutions. Usage of the developed pipeline library elements reduced using of the logic blocks. Usually, the number of logic blocks in FPGA chip is less than the number of triggers. So, the chip XC7VX585T contains the logic blocks the number of which twice is less than the numbers of the triggers. Therefore, this resource is more important. Due to its reduction twice more computational devices managed to place into one FPGA in comparing with other implementations.

We have not investigated the comparison of hardware implementation of standard encryption DES algorithm of hardware description languages, because the design time of the solutions in the languages of hardware description is higher than the design time of the solutions in the high-level languages. Nevertheless, the solutions which were performed using HDL-groups languages are the most effective at the moment. It should be noted that the performance of the programming of applied task solutions using multichip RCS software complex is about 85-90% more effective in comparing with the solutions that are obtained using HDL-languages.

7 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 that provides high effectiveness of calculations and practically linear growth of performance of increasing computational resource.

Implemented tasks of speckle image (astronomical object identification) and data (standard encryption DES) processing with usage the developed RCS programming suit have 2.5-5 times higher real performance than the solutions that obtained with other programming tools.

It should also be noted that the developed programming suit is provided the multichip RCS programming in one project, without explicitly specifying the distribution of calculations on FPGA chips that is not supported in other programming tools.

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

References

[1] K. M. A. Abd El-Latif, H. F. A. Hamed, E. A. M. Hasaneen, FPGA Implementation of the Pipelined Data Encryption Standard (DES) Based on Variable Time Data Permutation, *The Online Journal on Electronics and Electrical Engineering (OJEEE)*, Vol. 2, No.3, 2007.

[2] S. Akella, *Porting EDIF Netlists to the Viva Environment for Integrated Custom Computing Applications*, Proceedings MAPLD-2003, Military Applications of Programmable Logic Devices.

[3] Ali M. Reza, Adaptive Noise Filtering of Image Sequences in Real Time, *WSEAS Transactions on Systems*, Vol.12, No.4, 2013.

[4] *Altium Designer FPGA Design Basics*, Altium Limited, 2008.

[5] Convey computer, http://www.conveycomputer.com

[6] Esam El-Araby 1, Mohamed Taher 1, Mohamed Abouellail 1, Tarek El-Ghazawi 1, B. Gregory, *Comparative analysis of high level*

programming for reconfigurable computers: methodology and empirical study, Proceedings of the 3rd Southern Conference on, SPL '07, Newby, Programmable Logic, pp. 99-106, 2007.

[7] Design for Xilinx FPGAs using high-level languages and Vivado HLS, *Journal Components and technologies*, No.12, 2013.

[8] I.A. Kalyaev, I.I. Levin, A.I. Dordopulo, E.A. Semernikov, *High-performance reconfigurable computing system of new generation*, Proceedings of the International supercomputer conference with elements of scientific school for youth "Scientific service in the Internet: exoflops future", MSU publishing, pp. 42-49, 2011.

[9] I.A. Kalyaev, I.I. Levin, E.A. Semernikov, V.I. Shmoilov, *Reconfigurable multipipeline computing structures*, Nova Science Publishers, 2012.

[10] I.A. Kalyaev, I.I. Levin, A.I. Dordopulo, L.M. Slasten, FPGA-based Reconfigurable Computer Systems, *Science and Information Conference (SAI)*, 2013, pp. 148-155.

[11] I.A. Kalyaev, I.I. Levin, A.I. Dordopulo, L.M. Slasten, *Reconfigurable Computer Systems Based on Virtex-6 and Virtex-7 FPGAs*, IFAC Proceedings Volumes (ISSN 14746670), Programmable Devices and Embedded Systems, Vol. 12, No.1, 2013, pp. 210-214.

[12] *Libero IDE v9.1 User's Guide*, Actel Corporation, 2010.

[13] Maxeler Technologies, http://www.maxeler.com/

[14] 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*, Vol.12, No.3, 2013.

[15] Nallatech, a subsidiary of Interconnect Systems Inc., http://www.nallatech.com/

[16] Picocomputing, http://picocomputing.com/

[17] *Quartus II Handbook Version 10.1 Volume 1: Design and Synthesis*, Altera Corporation, 2010.

[18] Raed Bani-Hani, Salah Harb, Khaldoon Mhaidat, Eyad Taqieddin, High-Throughput and Area-Efficient FPGA Implementations of Data Encryption Standard (DES), *Journal Circuits and Systems*, Vol.5, 2014, pp. 45-56.

[19] SRC computers, http://www.srccomp.com/

[20] D. Tamjidi, *An Overview and Benchmark Study of the Starbridge Reconfigurable Computing Platform*, UNIVERSITY OF CALIFORNIA, 2005.

[21] Y.Y. Zotov, *The design of digital devices based on FPGA by XILINX to WebPACK ISE* [Text], Moscow, Hot line-Telecom, pp. 624, 2003.