High Performance Reconfigurable Computer Systems on the base of FPGA technology

I.A. Kalyaev*. I.I. Levin**. L.M. Slasten***

*Kalyaev Scientific Research Institute of Multiprocessor Computer Systems of Southern Federal University, Taganrog, Russia, (Tel: +7-8634-360-376; e-mail: kaliaev@mvs.sfedu.ru).
**Kalyaev Scientific Research Institute of Multiprocessor Computer Systems of Southern Federal University, Taganrog, Russia, (Tel: +7-8634-319-092; e-mail: levin@mvs.tsure.ru).
***Southern Scientific Centre of the Russian Academy of Sciences, Rostov-on-Don, Russia, (Tel: +7-8634-318-910; e-mail: lmslasten@yandex.ru).

Abstract: The paper covers the problems of design and creation of high performance reconfigurable computer systems based on FPGA technology. Authors compare performance values of up-to-date cluster systems and original reconfigurable computer systems, suggest principles of design of modular scalable reconfigurable computer systems with high real performance and give examples of various implementations of such systems. In addition, principles of applications development are suggested. Also the paper covers the problems of reconfigurable system programming. There is viewed an original complex of system software for reconfigurable system programming.

Keywords: multiprocessor systems, supercomputers, reconfigurable system architecture, high performance, parallel processing, pipeline processing, computer-aided circuit design, structure-oriented languages, interfaces, hardware.

1. INTRODUCTION

Until recently supercomputer technology had been developed according to large-block integration principles. Performance was provided due to technological achievements in the field of microprocessor design and communication systems development, but advantages of architecture and circuit design were not used. As a result, multiprocessor systems of cluster type designed on the base of widely-available computer nodes and communication networks became widespread during the last years. Cluster systems have relatively low price and simple programming methods. Software developed and debugged on personal computer may be rather easily adapted to such systems. However, cluster supercomputers generally provide high real performance only at solving loosely-coupled tasks when a number of data exchange operations is rather low. But if the solving tasks are tightly-coupled, performance of cluster systems considerably reduces and does not exceed 5-10% from the declared peak performance of the system [Aladyshev, Dikarev, Ovsyannikov, 2004]. Besides, cluster system performance reduces when a number of processors in cluster supercomputer grows. This is the result of dissimilarity between task information structure and cluster supercomputer “fixed” architecture. Real performance of supercomputer decreases and burden sharply increases if the solving task is tightly-coupled and requires a great number of data exchange operations. The concept of multiprocessor computer systems with reconfigurable architecture had proved successful in solving specified problems of cluster supercomputers [Kalyaev and Levin, 2003; Kalyaev, Levin, Semernikov, Shmoilov, 2009; Dmitrenko, Kalyaev, Levin, Semernikov, 2009].

The essence of the concept is adaptation of reconfigurable computer system (RCS) architecture to the structure of information graph of the solving task. Usage of the principles of the concept can significantly reduce computational burden and increase system real performance.

2. DESIGN PRINCIPLES OF RECONFIGURABLE COMPUTER SYSTEMS BASED ON FPGA FIELDS

Fixed unchangeable architecture is the cause of low real performance at solving specific problems. While mapping of a real task on such fixed architecture, large nonproductive time costs originate. The reason of the time costs is organization of computational process, but not performing of useful calculations.

The main idea of reconfigurable computer systems creation is to give application programmer an opportunity to adapt computer architecture to the structure of the solving task. Time needed for computational process organisation is minimized and hence real performance is increasing.

Somehow, the idea of adaptation of computer system architecture to the solving task was used in homogeneous computing media [Evreinov, Khoroshevsky, 1978], systolic structures [Moore, McCabe, Urquhart, 1987], systems with reconfigurable (programmable) architecture [http://www.dinigroup.com; http://www.copacobana.org; http://www.sciengines.com]. However, in spite of numerous and various research several prototypes of these systems were designed, but the matter did not get any farther due to...
lack of element base satisfying requirements of the concept of reconfiguring of computer system according to the task structure.

The necessary element base is field programmable gate arrays (FPGA) of large-scale integration designed in the beginning of the XXI century. In contrast with standard integrated circuits with fixed internal structure defined on manufacturing stage, FPGA internal structure is programmable and may be varied by circuit engineer during operation. Consequently, FPGA may be used as reconfigurable co-processors for standard microprocessors. While carrying out nonstandard computing operation, a computing structure for hardware realisation of the operation is created in FPGA. The examples of using FPGAs as co-processors for standard microprocessors are the following: supercomputer CRAY-XT5h [http://www.cray.com] and supercomputer SGI Altix 4700 (35 microprocessor computing modules with two FPGAs in each module) manufactured by Silicon Graphics Inc. [http://www.sgi.com].

Nevertheless, using FPGA as co-processors is ineffective, because a single FPGA may contain computing structures providing several operations. Reprogramming of FPGA for each subsequent operation also needs time and leads to reducing of system real performance during the task solving.

FPGA may be considered as a component of a computational field within which multipipeline computing structures, realizing the entire graph of the algorithm, may be created. This concept provides fundamentally new opportunities of FPGA usage. The task is parallelized in the FPGA field according to the principle of fine-grain parallelization, time needed for computational process organization is minimized and hence high technical parameters of computer system, such as ratios “real performance/peak performance”, “real performance/power consumption” and “real performance/volume”, are provided.

Given below are the main principles of organisation and functioning of RCS based on FPGA computational fields.

1. Some FPGAs are combined into a computational field (Fig. 1).

2. A multipipeline computational structure, similar to the informational graph of the solving task, is designed within the field of FPGA using computer-aided design and structural programming software tools (Fig. 2).

3. If the informational graph of the solving task can not be mapped on the FPGA field because of hardware constraints, it must be split into disjoint subgraphs. Each subgraph must be mapped on the FPGA field, i.e. structurally realized. Sequential procedure of subgraphs structural realization within the FPGA field is organized after mapping (Fig.3).
It is evident that the larger is the FPGA field, the smaller number of subgraphs is obtained after the informational graph splitting. As a result, some burden in the FPGA field reconfiguring may be reduced and RCS real performance at the task solving may be increased.

RCS, designed for solving complicated problems, has to contain hundreds and thousands of FPGAs of large-scale integration, combined in the computational field. It is clear, that placing of such number of FPGAs on the one circuit board is impossible. This problem can be solved using a principle of modular design of the RCS computational field on the base of unified basic modules. Basic module is a board which contains a fragment of the FPGA computational field and auxiliary elements: intermodule data exchange interfaces, distributed memory blocks, secondary power units, synchronization subsystem, control nodes, network interfaces, etc. Basic module is a small-sized RCS which is able, along with a personal computer, to solve user's tasks. Several basic modules may be united into an RCS with the needed performance. Generalized structure of the RCS basic module is given in Figure 4.

The main computational abilities of the basic module are concentrated in its computational field which contains a number of large-scale integration FPGAs. Distributed memory blocks are realized on the base of RAM standard chips SRAM or SDRAM of the needed capacity and transfer rate. Control of the basic module resources is realized by basic module controller (BMC). Such procedures as initial data loading and result data unloading, loading fragments of parallel applied programs into distributed memory controllers are also realized by BMC.

FPGA connections within the computational field are to provide high data transfer rate between parts of computational structures placed in other microchips of the computational structure multichip implementation. Therefore connections between FPGAs have a minimum length and are implemented using LVDS (Low Voltage Differential Signaling) or RocketIO technologies.

3. EXAMPLES OF RCS IMPLEMENTATION

Some types of the RCS basic modules (BM), designed in Kalyaev Scientific Research Institute of multiprocessor computer systems of Southern Federal University (SRI MCS SFU) and Scientific Research Centre of Supercomputers and Neurocomputers (Taganrog, Russia), are given in Figure 5.

A family of RCSs of various performance – from 50 GFlops (16 FPGAs Virtex 5) to 6 TFlops (1280 FPGAs Virtex 5) – was designed on the base of BM 16V5-75 (Fig. 6) [Dmitrenko, Kalyaev, Levin, Semernikov, 2009; Kalyaev, Levin, Semernikov, 2010].

Fig. 5. RCS basic modules

Fig. 6. The family of RCS on the base of the FPGA computational fields

RCS-5 contains five 19’ ST-1R racks. RCS-1R contains one 19’ ST-1R rack. Each rack contains four blocks RCS-0.2-CB designed according to “Euromechanics” 6U standard. Each block contains up to four basic modules 16V5-75 with peak performance up to 200 GFlops each for single precision data. Connections within the computational field 16V5-75 are implemented according to LVDS standard and provide total...
data transfer rate over 3 TBit/sec at a frequency of 1200 MHz.

At present specialists of SRI MCS SFU work on the design of modular scalable RCS with performance about 20 TFlops. The RCS will be placed in 19’ rack and will contain 1536 FPGAs Virtex 5. The RCS has been designed on the base of the BM 16V5-1250R with performance about 250 GFlops. High speed LVDS channels connect all basic modules into a unified computational resource.

Four basic modules 16V5-1250R are placed in a block “Orion”, designed according to “Euromechanics” 1U standard (Fig. 7). All FPGA chips of the computational field of the basic module 16V5-1250R have LVDS channels for computational field expansion in contrast to the basic module 16V5-75 which has only four of sixteen chips with expansion ability. This distinction provides high data exchange rate between basic modules of the block “Orion”.

![Fig. 7. Prototype of the RCS computational block “Orion”](image)

Real performance values of the blocks RCS-0.2-CB and “Orion” at executing tasks of various classes, such as digital signals processing, linear algebra and mathematical physics, are given in the Table 1.

**Table 1. Performance of the blocks RCS-0.2-CB and “Orion”**

<table>
<thead>
<tr>
<th>Tasks</th>
<th>DSP, GFlops</th>
<th>Linear algebra, GFlops</th>
<th>Mathematical physics, GFlops</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCS-0.2-CB</td>
<td>647.7</td>
<td>423.2</td>
<td>535.2</td>
</tr>
<tr>
<td>Orion</td>
<td>809.6</td>
<td>528.7</td>
<td>669.0</td>
</tr>
</tbody>
</table>

Values of the Table 1 show that the computational blocks “Orion” and RCS-0.2-CB have almost the same computational field resources, but the block “Orion” has higher performance. This is explained by high capacity of the channels, connecting computational fields of basic modules, and by larger number of distributed memory channels connected to the computational field: 80 channels in the block RCS-0.2-CB and 128 channels in the block “Orion”.

Values of performance per unit volume for these two blocks are rather different. For the block “Orion” this value is higher because it has more rational design. Volume of “Orion” is 18% from total volume of the block RCS-0.2-CB or 32% from its computational part. Table 2 contains values of specific performance of the block RCS-0.2-CB and the block “Orion” at solving tasks using floating point single precision data. High specific performance of the block “Orion” is very important for design of multirack high performance RCSs.

**Table 2. Values of blocks specific performance**

<table>
<thead>
<tr>
<th>Performance</th>
<th>General tasks (GFlops/dm³)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCS-0.2-CB</td>
<td>16.7</td>
</tr>
<tr>
<td>Orion</td>
<td>64.9</td>
</tr>
</tbody>
</table>

The block “Orion” and the rack ST-1R have no external high rate LVDS-channels for resource expansion of the computational fields. Performance of the systems which consist of the blocks “Orion” and the racks ST-1R may be increased by using network technologies via Ethernet channels, as it is made in RCS-5 [Kalyaev, Levin, Semernikov, Shmoilov, 2009; Dmitrenko, Kalyaev, Levin, Semernikov, 2009; Kalyaev, Levin, Semernikov, 2010].

4. RECONFIGURABLE COMPUTER SYSTEMS PROGRAMMING

Process of software development for RCS greatly differs from the one for multiprocessor computer systems with cluster architecture. Software development for RCS may be divided into two parts: structural programming and procedural programming. The results of structural programming are computational structures within the field of FPGA logical cells, necessary for calculations. The sense of procedural programming is very similar to traditional programming and consists in creation of computational process in RCS. Structural programming of the FPGA computational field is the most difficult for RCS programmer, because traditional programming consists only in creating computational process based on fixed hardware. For programming of computational structures, based on FPGA field, programmer needs quite different skills – skills of a circuit engineer.

A special software complex for applied programs development helps to make RCS programming more simple. Using this complex user can develop applications without any special knowledge in the field of FPGA circuit design [Kalyaev and Levin, 2003; Kalyaev, Levin, Semernikov, Shmoilov, 2009; Levin, 2000]. The software complex allows:

- both structural and procedural programming based on high level programming language COLAMO;
- development and modification of computational structures for applied tasks without involving high qualified circuit engineer;
- development of portable applications for RCS with various architectures;
- applications scaling at resource expansion;
- remote using and monitoring of RCS computational resources.

According to functions the software complex may be divided into two parts:
- tools for applications development;
- tools for RCS computational resource control and administration.

The tools for applications development contain: integrated development environment Argus IDE with support of Argus and COLAMO programming languages; translator of RCS high level programming language COLAMO; translator of Argus assembler language; development environment of circuit solutions Fire!Constructor for synthesis of scalable parallel-pipeline structures which is based on IP-cores library (library of computational structures and interfaces).

A general structure of RCS application development is given in Figure 8.

Fig. 8. General structure of RCS application development

At the beginning programmer develops a parallel program (application) using high level programming language COLAMO and integrated development environment Argus IDE. Then the parallel program is translated by COLAMO-translator which generates four components of parallel application: control, procedural, stream and structural.

Control component is translated into Pascal language. Like control component, both of these components are elements of parallel program. Procedural and stream components are executed by distributed memory controllers and provide parallel data flows in computational structures.

Further, on the base of parallel program, COLAMO-translator generates object form of structural component, which describes informational graph of calculations. Using this object form, the development environment of scalable circuit solutions Fire!Constructor synthesizes computational structures for all FPGAs of the computational field. Synthesis and modifying of computational structures, generated for the parallel program, is executed automatically, without assistance of high qualified circuit engineer. In addition, parallel program is portable and may be executed on RCS of various architectures due to using libraries of descriptions of basic modules, blocks, RCSs (RCS passport). Initial data for Fire!Constructor is the following:

− structural component of COLAMO-application in object form, generated by the COLAMO-translator;
− library of computational structures and interfaces (IP-cores), used for informational graph mapping on RCS hardware. IP-cores of computational structures and interfaces are designed by circuit engineers during RCS design and creation. Then VHDL-descriptions of IP-cores are included into the library which is used by the Fire!Constructor;
− library of descriptions of basic modules, blocks and RCS (RCS passport). Using the RSC passport library, Fire! Constructor generates platform-independent multichip structures within the FPGA computational fields of basic modules, blocks and RCS racks, including heterogeneous RCSs, which consist of various basic modules and blocks.

Fire!Constructor automatically synchronizes data flows within a multichip implementation of parallel-pipeline computational structure according to RCS computational fields features: number and types of connections between FPGAs of basic modules and connections between computational fields of basic modules, blocks and racks. As a result, Fire!Constructor generates the following information for each FPGA of the RCS computational field:

− VHDL-description (*.vhd file) of computational structure fragment within the FPGA;
− information of correspondence (*.ucf file) between external signals logical names of computational structure fragment and pins of the FPGA.

The final step of parallel program development is configuration files creation. Xilinx ISE, using result data obtained from Fire!Constructor, generates configuration files (*.bit files) for further loading into FPGAs of the computational field.

For parallel program execution all configuration files must be loaded into FPGAs of the computational field and all...
components of the parallel application must be loaded into control processors and distributed memory controllers.

The software complex provides development of effective applications for RCS for solving tasks from various problem domains, simple programming and automatic porting of structural solutions from one RCS architecture into another. Usage of software complex reduces cost (in 2-3 times) and time of application development (in 3-5 times) in comparison with traditional method, when RCS computational structure is designed by a circuit engineer.

5. CONCLUSIONS

FPGA-based reconfigurable computer systems are fundamentally new direction of high performance technique development. In contrast to cluster supercomputers, RCSs allow to create, within basic architecture, virtual special-purpose calculators with structure similar to the solving task structure. This provides high effectiveness of calculations and almost linear performance growth at computational resource expansion.

General-purpose architecture of RCSs on the base of FPGA computational fields, which can be reprogrammed according to the structure of the solving task, offer unique opportunities of using RCS in problem domains which require:
- highly effective computer facilities, like special-purpose computer systems;
- ability of solving tasks from various problem domains.

RCSs with general-purpose basic architecture and ability of architecture reconfiguring entirely satisfy these requirements.

REFERENCES


http://www.dinigroup.com
http://www.copacobana.org
http://www.sciengines.com
http://www.cray.com
http://www.sgi.com
