Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC-PapersOnLine 49-25 (2016) 372–377

Parallel Application Porting by Means of Soft-Architectures

A.I. Dordopulo*.V.A. Gudkov*. A.A. Gulenok*.
V.B. Kovalenko. L.M. Slasten*. A.V. Pelipets*. A.V. Chkan**.**

**Scientific Research Institute of Multiprocessor Computer Systems at Southern Federal University, Taganrog, Russia, (e-mail: scorpio@mvs.tsure.ru).*
***Scientific Research Centre of Supercomputers and Neurocomputers, Taganrog, Russia, (e-mail: chkan_andrey@mail.ru).*

Abstract: In the paper we consider a method of porting of parallel applications to various reconfigurable computer systems with different soft-architectures and configurations. Our method of porting of parallel applications by means of soft-architectures contains a procedure of analysis and comparison of objects of the initial and the target soft-architectures, a procedure of analysis and comparison of connections of the ported parallel application and the target soft-architecture, a procedure which generates recommendations for modification of the target soft-architecture for successful porting. Together these procedures provide porting of the application to the target soft-architecture without additional modification and debugging of the application, and if porting is impossible the developer gets recommendations for modification of the target soft-architecture. Owing to this, it is possible to port the application, to simplify modification of the target soft-architecture and to reduce considerably the porting time. Owing to our new porting method, the parallel application is ported to a new soft-architecture without any additional modification of its source code, and its information graph is automatically mapped on various soft-architectures.

Keywords: reconfigurable architecture, programmable soft-architecture, FPGA, reconfigurable computer system, high performance, parallel processing, pipeline processing, computer-aided circuit design.

1. INTRODUCTION

Reconfigurable computer systems designed on basis of FPGAs provide adaptation of their own architecture to the structure of the solving task. Such systems are more and more applied for implementation of scientific and technical problems that become more and more complex. Growth of the number of FPGA gates leads to considerable (up to 3-5 days) increasing of the configuration translation time of chips, and hence, it considerably increases the debugging time of applications developed for multi-chip reconfigurable computer systems (RCS).

One of promising approaches of reduction of application debugging time is a concept of soft-architectures, suggested by I.I. Levin and E.A. Semernikov [Gudkov, Gulenok, Kovalenko, Slasten, 2013; Kalyaev, Levin, Dordopulo, Slasten, 2013]. Owing to this concept it is possible to reconfigure the task structure without FPGA chips re-programming that usually takes plenty of time.

An RCS soft-architecture (SA) is an architecture which allows creation of computational structures for tasks using only program adjustment and no FPGA re-programming.

If the developer works with soft-architectures, he himself can create and define types and interconnections of macro-objects, only according to features of solving tasks and requirements to the RCS real performance. A macro-object is a firmware workpiece which contains both circuit components and facilities for their programming, which provide re-adjustment without external control.

The main feature of SA use consists in no need of re-synthesis of bitstream files of a special-purpose SA when the source code of parallel applications is modified. Since synthesis of SA bitstream files takes a considerable part of the RCS programming time (75-90%), then absence of re-synthesis of bitstream files after modification of the source code of parallel applications leads to multiple reduction of the debugging time of complicated parallel-pipeline RCS applications.

Reduction of the debugging time of RCS applications is possible if we develop applications on the level of general-purpose macro-objects. Such macro-object is a set of computational blocks which perform a certain group of instructions and which are interconnected by a communication system. Macro-objects allow reconfiguration of the RCS structure during execution of the task. Here we develop applications using a low-level assembler oriented to a fixed RCS architecture and a fixed set of components (a family and type of FPGAs), therefore when we port an application, developed and debugged for one architecture, to an RSC with another architecture or configuration, we must considerably modify the source code of this parallel application. In fact, we must develop a new application for an RCS with a different architecture.

The base of the soft-architecture concept consists in use of specialized software tools for computational structure description, for computational structure synthesis, and for porting applications to various soft-architectures without modification of their source code.

10.1016/j.ifacol.2016.12.071

When we develop applications [Ke-Horng Chen, 2016; Schrainer, 2005; Geist et al., 1994] on the level of soft-architectures, sometimes it is necessary to port developed and debugged applications from one SA to another. The initial and the target SAs can contain different numbers of macro-objects with different features or RCS hardware resource. The natural way to solve the problem of porting is to modify the source code of the application, and then to translate the application and to debug it on the target SA. Such approach takes plenty of time. If the SA contains a large number of complex hierarchic macro-objects, then the developer must modify the application according to the target SA if he wants to port the application to this target SA. This process takes almost the same time as development of a new application. As a rule, it is necessary to port to the target SA not a single application, but several applications that were developed and debugged for some initial SA. In this case the required time can also be unacceptably long. To solve this problem we have created a method which provides porting of developed applications from some initial SA to some target SA without modification of their source code.

2. THE GENERALIZED METHOD OF PORTING OF PARALLEL APPLICATIONS

The concept of soft-architectures allows transformation of the task structure without re-programming of FPGA chips, which usually takes a long time. The basis of the soft-architecture concept (SA) is macro-objects. Each macro-object is a set of computational blocks performing a certain group of instructions and interconnected according to some communication system.

The developed method determines conditions and criteria of replacement of complex hierarchic macro-objects of the initial SA with macro-objects of the target SA with keeping information equivalency of performed operations. The distinctive feature of our porting method is search and comparison of macro-objects of various SA not according to their atomic characteristics but also according to a group of operations, set by operation patterns and used in the application or in the packet of applications. One of the principal steps of the method is preliminary verification of replacement of the SA operation elements taking into account their connections. The sense of such verification is functional comparison of macro-objects of different SAs, particularly comparison of operation patterns performed by these macro-objects. If porting is possible then each portable application will be mapped on the target SA by specialized synthesis software tools. If porting is impossible then a list of recommendations for the target SA modification is generated for each application. The recommendations will help the architect, who developed the target SA, to modify it for successful porting of the whole package of applications.

SAs are compared taking into account placement of the information graph of the portable task (application) on the initial SA. Owing to this, we can analyse only those components of the SA, which are used for implementation of the task, and we can avoid complete analysis of the initial SA, which contains analysis of all unused components of the initial SA and can take rather long time. As a result, selective comparison can reduce the porting time.

In general case a SA is a description of objects and connections between them, and functional features that depend on the type of the object. If we want to port the application, developed for the soft-architecture SA1, to the target soft-architecture SA2, we must take into account the features of SA1 and SA2, and the computational structure of the task, represented as the information graph.

The necessary conditions for possible porting between the initial SA1 and the target SA2 are:

- the “Memory”-type objects of the initial SA1 can be mapped on the “Memory”-type objects of the target SA2, and their functions are the same (“read” and “write” functions);

- any object except the “Memory”-object must be mapped on an object or a group of objects of the target SA2;

- the number of objects used by the application cannot exceed the number of objects of the target SA2.

The principal steps of the method of porting of parallel applications from the initial SA1 to the target SA2 are:

1) analysis of the objects of the initial SA1;

2) analysis of the objects of the target SA2;

3) analysis of the objects of the information graph of the task (application), which belongs to the tasks (applications) package;

4) association of “Memory”-objects of the initial SA1 and of the target SA2;

5) association of the objects of the initial SA1 and of the target SA2;

6) comparison of the number of specialized objects of the target SA2 and of the application;

7) modification of the objects of the task information graph according to the objects from the table of object corresponding;

8) generation of a warning that modification of the target SA2 is required. The warning contains information about the objects which were not replaced during modification of the objects of the information graph.

The developed method, which consists in association of objects of different SAs and in modification of the objects of the task information graph of the target SA, allows porting of the application to various SAs without modification of the application source code, and, hence, provides considerable reduction of the application development time.

3. ANALYSIS AND COMPARISON OF THE OBJECTS OF THE INITIAL AND THE TARGET SOFT-ARCHITECTURES

The initial data for the procedure of analysis and comparison of the objects of the initial and target SAs are files from the library of portable applications, which contain information of application mapping on the initial SA. These files also contain information about correspondence between the SA objects, placed on the specific hardware platform, and the vertices of the task information graph. Besides, we must take into account that each application can be a multi-cadr one. In