Multi-controller reconfiguration system for FPGAs

Chiraz Trabelsi*, Samy Meftali* and Jean-Luc Dekeyser*
*INRIA Lille Nord Europe - LIFL - USTL - CNRS, 40 avenue Halley, 59650 Villeneuve d’Ascq, FRANCE
Email:{Firstname.Lastname}@inria.fr

I. INTRODUCTION
Adaptivity is one of the most critical issues related to System-on-Chip (SoC) design. In order to be runtime adaptive, SoC have to take into account changes related to user preferences and environment at runtime. Dynamically reconfigurable SoC, such as those implemented on Field Programmable Gate Arrays (FPGAs), are a good solution for runtime adaptivity. Dynamic reconfiguration allows FPGAs to modify their functionality at runtime. Partial Dynamic Reconfiguration (PDR) has the potential of reconfiguring only some regions of the FPGA without disturbing the operation of the rest of the system, which decreases the impact of the reconfiguration time on the system performance, compared to the full reconfiguration. As the complexity of the reconfigurable SoC is increasing day after day, their reconfiguration control becomes more and more complex. In this case, using a single controller that controls the whole system might lead to a very complex design which is difficult to explore and to test. In this paper, we propose an approach that aims to decrease this complexity by dividing it into sub-modules. This approach is based on a multi-controller solution for the reconfiguration control of a partially reconfigurable system. Each controller controls the reconfiguration of a part of the system. Communicating together, the controllers insure the reconfiguration control of the whole system.

II. RELATED WORKS
Partial Dynamic Reconfiguration (PDR) improves the efficiency of the runtime reconfiguration, since it allows for the reconfiguration of only some of the logic gates, and leave the other ones unchanged. It is thus faster than the full dynamic reconfiguration. PDR can be applied to the software and hardware levels. At the software level, it can modify, for instance, the application code by reconfiguring the local memory of a processor [1]. At the hardware level, if an application task is allocated to a hardware accelerator, PDR allows the modification of this task by replacing the hardware accelerator by another [2]. Another hardware reconfiguration type is related to Reconfigurable Instruction Set Processors (RISPs), which contain a reconfigurable logic in one or more of their functional units. These processors can optimize their instruction sets according to the applications being under execution on them, by reconfiguring their hardware on fly [3]. At the communication level, communication primitives can be placed and routed dynamically. In [4], the authors propose a method to adapt signal routing to power and performance, taking into account the consumption of the different FPGA signal lines and their performances.

Reconfiguration Control can be distributed or centralized. Many works have used centralized controller [1] [2] [6]. For a complex reconfiguration control, which needs complex algorithms, distributed control is more efficient. In [5], the authors present a Mechatronic Control System in which configuration management and controller functionality are implemented directly in hardware. The implementation of the adaptive control system followed the Xilinx modular design flow. Each module and its corresponding tasks included a part of the distributed reconfiguration and activation control. This solution reduced the global reconfiguration management significantly compared to the centralized hardware reconfiguration controller used in [6]. In [7], the distributed control was used for the reconfiguration of an organic computing system. The studied system was composed of hardware computing cells implemented on FPGA. Each cell provides services such as computing, memory and I/O, and contains also a configuration control unit implemented on hardware. All the cells can communicate with the ICAP in an exclusive way, by establishing virtual channels. This solution allowed to accelerate the reconfiguration process compared to a centralized access to the ICAP.

Objectives and contributions
The main originalities of our approach are:
• distribution of the dynamic reconfiguration control between controllers, insuring each the control of a reconfigurable region. This distribution includes the monitoring, the reconfiguration decision-making process and the decision notification.
• use of negotiations between the controllers of the partially reconfigurable regions of the system.

III. RECONFIGURATION CONTROL DISTRIBUTION
In our approach for the distributed reconfiguration control, each controller controls the reconfiguration of a reconfigurable region. Figure 1 shows an example of a system containing two reconfigurable regions controlled by two controllers. The control distribution is based on the distribution of the monitoring of the reconfigurable regions, the decision-making process about reconfiguration, and the notification of the rest of the system when a reconfiguration decision is made. In Figure 1, the controllers are connected between the reconfigurable regions and the processor in order to monitor the exchanges between them.
configuration 1.1
configuration 1.m
Processor
can have to be reconfigured in order to insure a coherent system
of the other regions. In this case, the concerned regions
about its decision, because this might affect the configurations
ration of the controlled region, it notifies the other controllers
C. Notification
triggered by events detected through the monitored data.

B. Decision
Processing the monitored data, the controller can make a
decision about the reconfiguration of its controlled region. The
decision-making process is based on a Finite State Machine
(FSM). The states of this FSM are the possible configurations
of the controlled region. The transitions between the states are
triggered by events detected through the monitored data.

C. Notification
When the controller makes a decision about the reconfigu-
ration of the controlled region, it notifies the other controllers
about its decision, because this might affect the configurations
of the other regions. In this case, the concerned regions
have to be reconfigured in order to insure a coherent system
configuration.

IV. Configuration negotiation
Provided that the controller makes decisions that depend
only on a local vision of the system, the decisions might be
not suitable for the global configuration. Therefore, these
are not final decisions and have to be negotiated between
the controllers. The negotiations between the controllers are
managed by a configuration coordinator. Figure 2 illustrates
the coordination process. When a controller estimates that its
controlled region has to be reconfigured, it sends a reconfig-
uration request to the coordinator. According to this request,
the coordinator determines which other regions have to be
reconfigured in order to adapt to the request. To the concerned
controllers, it sends reconfiguration requests. The controllers
can either accept or refuse the requests. The reconfiguration
is authorized by the coordinator only if it receives positive
answers from all the concerned controllers. The reconfigu-
ration can be launched then via the Internal Configuration

V. CONCLUSION
In this paper, we presented our approach for a distributed
reconfiguration control on FPGA. This approach is based on
the use of controllers, controlling each a reconfigurable region
of the system. The reconfiguration decisions are taken in a
distributed way, and are negotiated between the controllers
in order to insure a coherence between the configurations
of the reconfigurable regions. As future works, we plan to
validate our approach by applying it with several embedded
applications. We also intend to work on optimizing the con-
trol algorithms, and generating automatically the distributed
control system.

REFERENCES
harvard-based processing element tailored for partial dynamic reconfig-
urable architectures,” in 22nd IEEE International Parallel and Distributed
Processing Symposium- 15th Reconfigurable Architectures Workshop,
2008.
time reconfigurable hardware to accelerate video processing in driver as-
sistance system,” in Proceedings of the conference on Design, automation
and test in Europe, 2007.
set processor design: Rt-risp,” in IEEE International Conference on
Computes Control and Communication, 2009.
power- dissipation by exploiting run-time adaption of communication
mechatronic control system supporting partial reconfiguration of con-
support for dynamic reconfiguration in reconfigurable soc architectures,”
in Field Programmable Logic and Application, 2004.
2d reconfiguration on xilinx virtex-fpgas for organic computing,” Int. J.