

AN FPGA-BASED DYNAMICALLY RECONFIGURABLE PLATFORM: FROM CONCEPT TO REALIZATION

Mateusz Majer

Department of Computer Science 12
University of Erlangen-Nuremberg
Erlangen, Germany
email: mateusz@cs.fau.de

ABSTRACT

Dynamically reconfigurable FPGA-based systems offer a new kind of flexibility such as on-demand computing, self-adaptation and self-optimization capabilities by restructuring the hardware at run-time. Using partial dynamic reconfiguration allows the main system to run uninterrupted during the reconfiguration process in addition to the reduced time for the reconfiguration process. However, existing FPGA-based platforms are hampered by physical restrictions limiting the practicability of partial reconfiguration. This led us to the concept of the *Erlangen Slot Machine* architecture in order to eliminate the physical and technical constraints.

1. INTRODUCTION

Until now, no FPGA-based platform provided a solution to the problems of design automation for dynamically hardware modules and their fast relocation. Advanced on-line scheduling and placement algorithms for future reconfigurable systems assume partial module reconfiguration on a 2D reconfigurable array and require a free space manager for the reconfigurable device which is coupled with a module database. However, current FPGAs are limited to 1D partial reconfiguration (slot-based with varying slot width). Furthermore, their partial abilities are physically restricted.

2. THE ERLANGEN SLOT MACHINE

The main idea of the Erlangen Slot Machine [1, 2] architecture is to accelerate the application development as well as the research in the area of partially reconfigurable hardware. The advantage of the ESM platform is its unique slot-based (varying slot width) architecture which allows the slots to be used independently of each other by delivering peripheral data through a separate crossbar switch as shown in Figure 1. The external crossbar performs the I/O pin virtualization for modules placed in slots. The ESM architecture is based on

flexible decoupling of the FPGA I/O pins from a direct connection to an interface chip. Moreover, inter-module communication infrastructure is paired with an automated tool support. This flexibility allows to place application modules at run-time in any available slot regardless of any I/O requirements. Thereby, run-time placement is not constraint through physical I/O pin locations as the I/O pin routing is performed through the programmable crossbar. Thus, the I/O pin dilemma is solved through dedicated hardware.

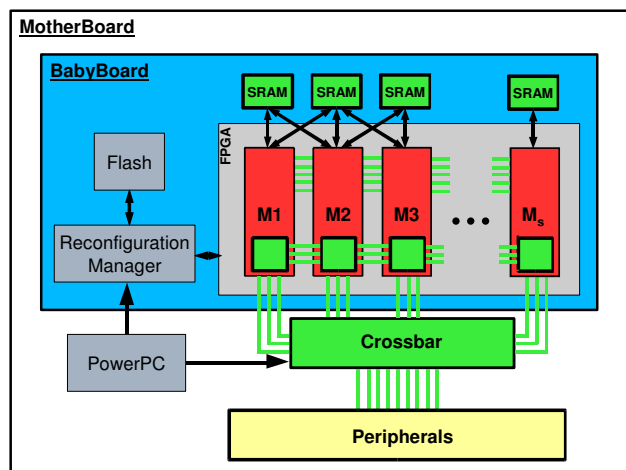


Fig. 1. ESM architecture overview shows the two board system. The main FPGA is located on the BabyBoard and is connected to a dedicated reconfiguration manager.

The ESM platform is centered around an FPGA serving as the main reconfigurable engine and an FPGA realizing the crossbar switch. The whole platform is separated into two physical boards called BabyBoard and MotherBoard and contains as main FPGA a Xilinx Virtex-II 6000 and as crossbar a Xilinx Spartan-IIE 600. Figure 1 shows the slot-based architecture of the ESM consisting of the Virtex-II FPGA, local SRAM memories, configuration memory and the reconfigu-

ration manager. The top pins in the north of the FPGA connect to local SRAM banks. These SRAM banks solve the problem of restricted intra-module memory in case of video applications. The bottom pins in the south are connected to the crossbar switch. Therefore, a module can be placed in any free slot and have its own peripheral I/O links together with dedicated local external memory.

3. INTER-MODULE COMMUNICATION

One of the central limiting factors, besides the I/O pin dilemma, for the wide use of partial dynamic reconfiguration yet not addressed is the problem of inter-module communication. Each module that is placed on one or more slots on the device must be able to communicate with other modules. For the ESM, we provide four main paradigms for communication among different modules: The first one is a direct communication using bus-macros between adjacently placed modules. Secondly, shared memory communication using SRAMs or BlockRAMs is possible. However, only adjacent modules can use these two communication modes. For modules placed in non-adjacent slots, we provide a dynamic signal switching communication architecture called Reconfigurable Multiple Bus (RMB) [3]. Finally, the communication between two different modules can also be realized through the external crossbar.

3.1. Communication between adjacent modules

On the ESM, bus-macros are used to realize a direct communication between adjacently placed modules, providing fixed communication channels that help to keep the signal integrity upon reconfiguration.

3.2. Communication via Shared Memory

This is particular useful in applications in which each module must process a large amount of data and then sends the processed data to the next module, as it is the case in video streaming. The six external SRAM banks are ideally suited for this approach.

3.3. Communication via RMB

In its basic definition, the RMB architecture consists of a set of processing elements or modules, each possessing an access to a set of switched bus connections to other processing elements. The switches are controlled by connection requests between individual modules. The RMB is a 1D arrangement of switches between N slots. In our FPGA implementation, the horizontal arrangement of parallel switched bus line segments allows for the communication among modules placed in the individual slots. The request for a new connection is done in a wormhole fashion, where the sender

(a module in slot M_k) sends a request for communication to its neighbor (slot M_{k+1}) in the direction of the receiver. Slot M_{k+1} sends the request to slot M_{k+2} , etc., until the receiver returns an acknowledgment. The acknowledgment is then sent back in the same way to the sender. Each module that receives an acknowledgment sets its switch to connect two line segments. Upon receiving the acknowledgment, the sender can start the communication (circuit routing). The wired and latency-free connection is then active until an explicit release signal is issued by the sender module.

4. CONCLUSIONS

A first prototype of the Erlangen Slot Machine was built and the concepts of I/O virtualization, inter-module communication for partial modules were successfully tested with a video-filter application. Currently we are working on a hardware reconfiguration manager [4] which will support on-the-fly bitstream relocation and manage the partial bitstream database. The PowerPC on the MotherBoard runs Linux and is the main controller for the whole platform. It will be used for on-line scheduling and placement of partial hardware modules on the main Virtex-II FPGA on the BabyBoard. We will also provide an API for the access to the hardware reconfiguration manager. Moreover, a tool called SlotComposer is under development which will allow an automated integration of application modules into the partial design flow. This is achieved through an automated wrapper generation for the desired inter-module communication schemes.

Further information and technical data sheets are available at <http://www.r-space.de>.

5. REFERENCES

- [1] C. Bobda, M. Majer, A. Ahmadiania, T. Haller, A. Linarth, J. Teich, S. Fekete, and J. van der Veen, "The Erlangen Slot Machine: A Highly Flexible FPGA-Based Reconfigurable Platform," in *IEEE Symp. on FPGAs and Custom Computing Machines (FCCM)*, Napa, U.S.A., 2005, pp. 319–320.
- [2] C. Bobda, M. Majer, A. Ahmadiania, T. Haller, A. Linarth, and J. Teich, "Increasing the Flexibility in FPGA-Based Reconfigurable Platforms: The Erlangen Slot Machine," in *IEEE Conference on Field-Programmable Technology (FPT)*, Singapore, 2005, pp. 37–42.
- [3] A. Ahmadiania, C. Bobda, J. Ding, M. Majer, J. Teich, S. Fekete, and J. van der Veen, "A Practical Approach for Circuit Routing on Dynamically Reconfigurable Devices," in *IEEE International Workshop on Rapid System Prototyping (RSP)*, Montreal, Canada, 2005, pp. 84–90.
- [4] M. Majer, A. Ahmadiania, C. Bobda, and J. Teich, "A Flexible Reconfiguration Manager for the Erlangen Slot Maschine," in *Dynamically Reconfigurable Systems Workshop (DRS)*, Frankfurt (Main), Germany, 2006, pp. 183–194.