A Multi–DSP Architecture with Dynamically Reconfigurable Coprocessors *

Erich Lind, Marco Platzner, and Bernhard Rinner
E-mail: rinner@iti.tu-graz.ac.at
Institute for Technical Informatics
Graz University of Technology

Abstract

We present the development of dynamically reconfigurable coprocessors that are embedded in a special-purpose computer architecture for the qualitative simulator QSIM. *The controller and the basic architecture for the reconfiguration are described and the overall reconfiguration time is evaluated.*

keywords:
reconfigurable computing devices, multi-DSP, FPGA, qualitative simulator QSIM

1 Introduction

Multiprocessor systems consisting of digital signal processors (DSPs) are used in applications with very high performance requirements [7][4]. These multi-DSP systems are sometimes complemented by application-specific coprocessors. Examples are coprocessors for accelerated execution of special functions [3] and coprocessors supporting data transfer and synchronization between processors and processors/memory modules [1].

For a class of applications the required coprocesssor functionalities vary strongly during runtime. As a consequence, the coprocessors are idle during a remarkable fraction of the overall runtime, leading to a poor efficiency. This problem gains importance for applications where the control-flow depends on the input-data. A concept to overcome this problem is to use dynamically reconfigurable coprocessors. The term *'dynamically'* denotes that the reconfiguration is done at runtime rather than at compile time. Dynamic reconfiguration is often implemented by SRAM-based FPGAs [2][6].

In [8] we have described the development of a computer architecture specialized for the qualitative simulator QSIM [5]. The performance of QSIM is improved by two strategies. First, the algorithm is parallelized and mapped onto a multi-DSP system. Second, frequently used functions are migrated from software to hardware and executed on specialized coprocessors. In this paper, we report on the development of dynamically reconfigurable coprocessors for this specialized QSIM computer architecture.

*This project is partially supported by the Austrian National Science Foundation *Fonds zur Förderung der wissenschaftlichen Forschung* under grant number P10411-MAT.

2 QSIM Coprocessors

The most frequently used functions in QSIM are the so-called *constraint-check-functions (CCFs)*. Like many block-oriented simulation paradigms, QSIM uses a set of constraints and a set of variables for system modeling. Examples for constraint types are $MULT$, ADD, and characteristic for qualitative simulation, M^+ and M^-. Typical operations of the CCFs are (i) evaluation of boolean expressions, (ii) comparisons, and (iii) set operations. As DSPs are not well suited for these kinds of operations, we have developed coprocessors for the most important constraint types [3]. These coprocessors have been implemented on Xilinx FPGAs of type XC4013.

A simulation run in QSIM consists of several subsequent simulation steps. In one simulation step, each combination of variables for each constraint in the simulation model is checked by a CCF. There are two cases where a dynamic reconfiguration of the coprocessors is desired. First, at the beginning of a simulation run the coprocessors should be reconfigured to match the distribution of constraint types in the currently simulated model. Second, as normally the number of available coprocessors is far less than the number of constraints, the coprocessors should also be reconfigured during a simulation step.

3 Dynamic Reconfiguration

3.1 Basic Architecture

Figure 1 displays the block diagram of the basic architecture for reconfiguration. The coprocessor is connected to a processor of the multi-DSP system via two independent unidirectional communication channels. The only additional device needed for the reconfiguration is the *configuration controller*. Like the coprocessor device, it is connected to a DSP via two communication channels. The configuration controller receives the configuration data from the multi-DSP system and reloads the coprocessor device. We use *pre-compiled* coprocessor designs, that are stored in the RAMs of the DSP system.

3.2 Configuration Controller

The communication channels between the DSP and the configuration controller transfer data in units of 32-bit words. The defined protocol differentiates between the command word, the configuration data words, and the status word.

To initiate a reconfiguration, the DSP sends a command word to the configuration controller. This word consists of an n-bit wide field selecting which of the at most n coprocessor devices has to be reconfigured and a field displaying the size of the reconfiguration data. The configuration controller activates the `PROGRAM` signal for the selected coprocessor. Then, the configuration controller successively receives a word of the configuration data from the DSP, converts it to a bit-stream, and transmits this bit-stream to the coprocessor device via the `DIN` line. The reconfiguration is done in the so-called *slave-serial* mode of Xilinx FPGAs which keeps the hardware effort minimal. The required clock signal, `CLK`, is provided by the configuration controller. If the coprocessor device detects a transmission failure, it activates the `INIT` signal; if the reconfiguration is completed successfully, it activates the `DONE` signal. These two signals are used by the configuration controller to generate the status word that is sent back to the DSP. This status word consists of a flag indicating the success of the reconfiguration and the number of words transmitted to the copro-

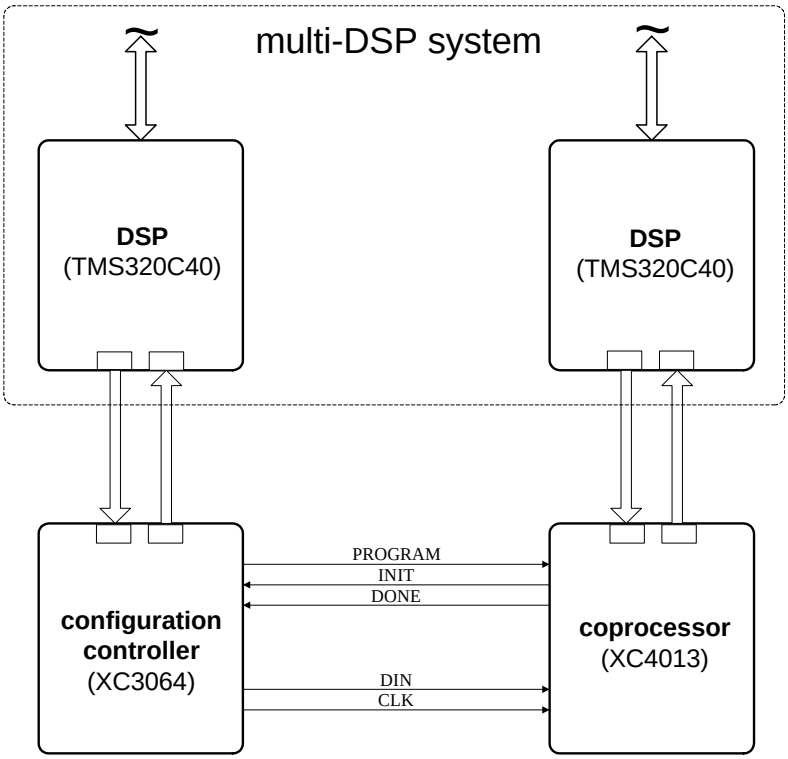

Figure 1: Block diagram of the basic architecture. The configuration controller and the coprocessor are connected to two DSPs of the multi-DSP system via two communication channels.

cessor device. In case of a failure, this information can be used for debugging purposes.

The configuration controller can be connected to at the most n coprocessor devices through its n `PROGRAM` lines. The `DIN` and `CLK` lines are shared by all devices, the `INIT` and `DONE` lines form a wired-OR. Each bit in the n-bit wide selection field of the command word corresponds to a coprocessor device. This allows the simultaneous configuration of up to n coprocessor devices with the same functionality.

3.3 Implementation and Evaluation

We have implemented a configuration controller with $n = 8$ in a Xilinx FPGA of type XC3064. The configuration clock frequency for the slave-serial mode is 5 MHz. The software on the multi-DSP system is running under the distributed multitasking operating system Virtuoso (Eonic Systems).

The reconfiguration process is initiated by a Virtuoso task. The end of the reconfiguration process is reached, when the DSP has received the status word from the configuration controller. The Virtuoso task measures the duration between start and end of the reconfiguration process by means of the TMS320C40's high-resolution timer. This duration is considered as *overall reconfiguration time*, including the time for the actual reconfiguration of the copro-

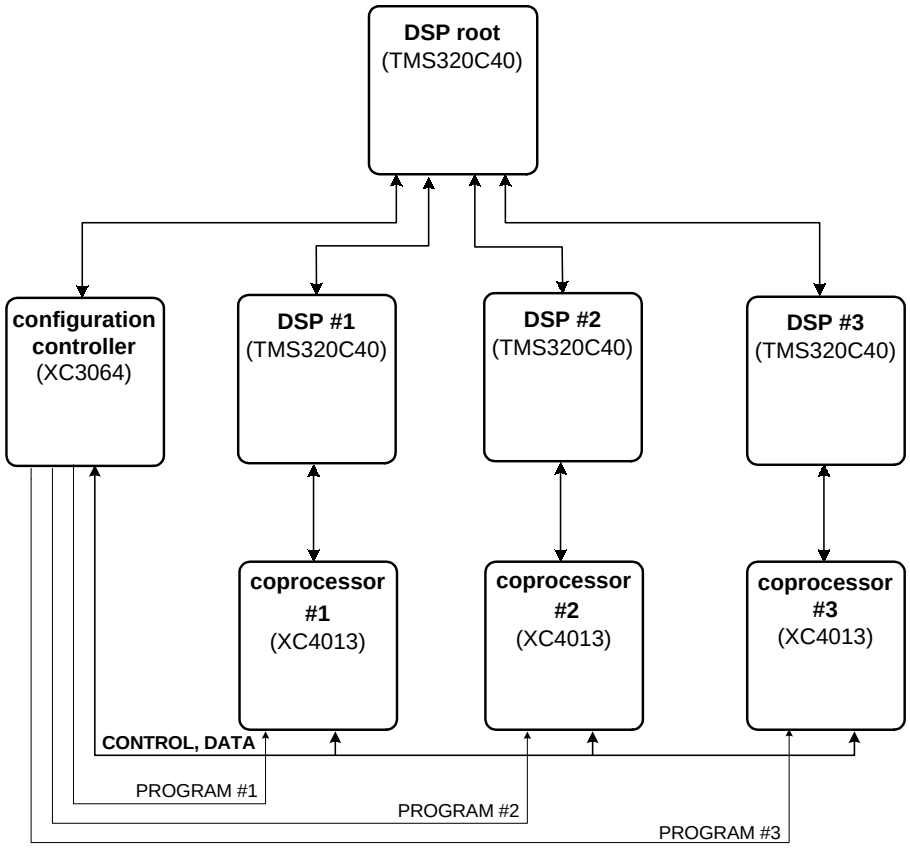

Figure 2: Overall QSIM computer architecture. The multi-DSP system consists of a *root*-DSP and 3 *leave*-DSPs. Each of the leave-DSPs has attached a specialized coprocessor. The configuration controller is connected to the root-DSP.

cessor device as well as the communication from the multi-DSP system to the configuration controller. In our experiments this time was 51 ms. The size of the configuration data for the XC4013 is 247960 bit.

4 Conclusion, Future Work

We have presented the development of dynamically reconfigurable coprocessors that are embedded into a multi-DSP architecture. The overall computer architecture is specialized for the simulation algorithm QSIM. The experiments have shown that a reconfiguration of the coprocessors is feasible from the reconfiguration time as well as from the size of reconfiguration data. Although varying in extreme cases, runtimes for one simulation step of the specialized QSIM architecture typically range from 10 ms up to 1300 ms [9] and sometimes more than 100 simulation steps are required for a complete simulation run. This means that the overall reconfiguration time is small enough to allow an adaption of the coprocessor functionalities at the beginning of a QSIM sim-

ulation run. However, for a reconfiguration of the coprocessors during one simulation step, much shorter reconfiguration times are necessary. This requirement is already reflected by new FPGA developments. For example, the announced Xilinx line XC62xx is optimized for fast reconfiguration. These FPGAs offer reconfiguration times of a factor up to 1000 shorter than current FPGAs and further also partial reconfiguration capabilities.

Ongoing and future work includes:

Optimizations. The reconfiguration controller and the prototype setup will be optimized concerning (i) a higher configuration clock and (ii) an improved error handling.

Integration. The QSIM computer architecture and the reconfiguration system will be integrated. Figure 2 shows an example of the overall computer architecture. Based on this architecture, an extensive evaluation of reconfiguration strategies can be done.

References

[1] E. Brenner and R. Weiss. A New Shared Memory Concept for Multiprocessor Systems with High-Speed Communication Links. In *Proc. ISCA International Conference on Parallel and Distributed Computing Systems*, pages 279–284, Orlando, USA, 1995.

[2] J. G. Eldredge and B. L. Hutchings. Density enhancement of a neural network using FPGAs and run-time reconfiguration. In *Proc. IEEE Workshop on FPGAs for Custom Computing Machines*, pages 180–188, Napa, USA, 1994.

[3] G. Friedl, M. Platzner, and B. Rinner. A Special–Purpose Coprocessor for Qualitative Simulation. In *Proc. First International EURO-PAR Conference*, pages 695–698, Stockholm, Sweden, 1995.

[4] R. Hranitzky and M. Platzner. Design and Implementation of Adaptive Digital Filters on a Multi–TMS320C40 System. In *Proc. International Conference on Signal Processing Applications & Technology*, Boston, 1995.

[5] B. Kuipers. *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge.* Artificial Intelligence. MIT Press, 1994.

[6] P. Lysaught and J. Dunlop. Dynamic reconfiguration of FPGAs. In *Proc. 1993 International Workshop on Field-Programmable Logic and Applications*, pages 82–94, Oxford, England, 1993.

[7] H. Matsumoto et al. Real-Time Simulator For Power System Using a Multi-DSP System (TMS320C40 x 32). In *Proc. International Conference on Signal Processing Applications & Technology*, pages 1338–1342, Boston, 1995.

[8] M. Platzner and B. Rinner. Improving Performance of the Qualitative Simulator QSIM — Design and Implementation of a Specialized Computer Architecture. In *Proc. ISCA International Conference on Parallel and Distributed Computing Systems*, pages 494–501, Orlando, USA, 1995.

[9] B. Rinner. *Design, Protoype Implementation and Experimental Evaluation of a Scalable Multiprocessor Architecture for Qualitative Simulation.* PhD thesis, Graz University of Technology, April 1996.