Optimal Mapping of the Numerical Simulation of Partial Differential Equations on Emulated Digital CNN-UM Architectures

Theses of the Ph.D. dissertation

Kiss András

Scientific Adviser
Dr. Péter Szolgay

Supervisor:
Dr. Zoltán Nagy

Péter Pázmány Catholic University
Faculty of Information Technology

Budapest, 2011

1. Introduction and aim

Due to the rapid evolution of computer technology the problems on many processing elements, which are arranged in geometric structures (array processors), become important. With the large number of the processor cores not only the speed of the cores but they topographic structure becomes an important property. These processors are capable to run multiple tasks in parallel. In order to make an efficiently executed algorithm, the relative distance between two neighboring processing elements should be take into consideration. In other words it is the phenomenon of the precedence of locality. This discipline requires the basic operations to be redesigned in order to work on these hardware architectures efficiently.

During my dissertation I am looking for solutions, where the area and dissipated power is minimal, the number of implemented processor, the speed and the memory access are maximal. During the search for the solution of the implementation of a partial differential equation solver I search within this parameter space, and I optimize the solution for some variable of this parameter space (e.g.: speed, area, bandwidth). The search will be always limited by the special properties of the hardware environment.

There are several known problems, which cannot be computed in real time with the former computing, just very slowly. The aim of my research is the examination of these hard problems, more precisely the investigation of the fluid flow simulation, and to make a hardware implementation for the problems. In the dissertation the methods will be investigated, which pro-

vides an opportunity to help solving these hard problems.

2.   Methods used in the experiments

The motivation of my dissertation is to develop a methodology for solving partial differential equations, especially for fluid and gas flow simulations, which helps to map these problems optimally into bounded and not bounded architectures. To reach this goal I investigated two hardware platforms, namely the IBM Cell Broadband Engine Architecture and the Xilinx Field Programmable Gate Array (FPGA) as reconfigurable architecture.

The IBM Cell processor represents a bounded architecture, which builds up from heterogeneous processor cores. From the marketing point of view, the Cell processor failed, but its significant innovations (e.g.: heterogeneous processor cores, ring bus structure) can be observed in todays modern processors (e.g.: IBM Power 7, Intel Sandy Bridge). According to the special requirement of the processor, I worked with vectorized data which composed of floating point numbers. For the development of the software I used the freely available IBM software development kit (SDK) with C programming language.

Xilinx FPGAs are belonging to the leading reconfigurable computers long ago. Due to the fast Configurable Logic Blocks (CLB) and to the large number of interconnections arbitrary circuits can be implemented on it. In order to accelerate certain operations dedicated elements (e.g.: digital signal processing (DSP) blocks) are available on the FPGA. The FPGA's CLB and DSP can be treated like different type of processors

which can handle different operations efficiently. Due to the
configurable parameters of the FPGA the processed data can
be represented in arbitrary type and size. During the research
I investigated fixed point and floating point numbers with dif-
ferent mantissa width in order to find the optimal precision for
a qualitative good result. During the implementation process
I used the Xilinx Foundation ISE softwares with VHDL lan-
guage. For the software simulation I used the MentorGraphics
Modelsim SE software.

3.   New scientific results

1. Thesis: *Development of an efficient mapping of the simulation of partial differential equations on inhomogenous and reconfigurable architectures: I have compared the optimal mapping of the simulation of a complex spatio-temporal dynamics on Xilinx Virtex FPGA and on IBM Cell architecture, and I made a framework for that. The framework has been successfully tested by the acceleration of a computational fluid dynamics (CFD) simulation. During the implementation my goal was always to reach the highest possible computational performance. The structure of the accelerator was designed according to this goal while considering the hardware specifications of the different architectures.*

1.1. **I have implemented an effective architecture, in the aspect of area, speed, dissipated power, bandwidth, for solving partial differential equations on structured grid. I have redesigned the arithmetic unit of the Falcon processor according to the discretized version of the partial differential equations optimized for the dedicated elements (BlockRAM, multiplier) of the FPGA.**

I have developed a process for the optimal bandwidth management between the processing elements and the memory on Xilinx Virtex and on IBM Cell architectures, which makes it possible to continuously supply the processing elements with data.

I have successfully confirmed experimentally in both cases, that placing a memory element close to the

processor results in a beneficial effect on the computing speed, which provides a minimum one order of magnitude higher speedup independently from the dimension of the problem.

1.2. **I have proved experimentally that one order of magnitude speedup can be achieved between an inhomogenous architecture, like the IBM Cell, and a custom architecture optimized for Xilinx Virtex FPGA using the same area, dissipated power and precision.** During the simulation of CFD on body fitted mesh geometry the Xilinx Virtex 5 SX240T running on 410 MHz is 8 times faster, than the IBM Cell architecture with 8 synergistic processing element running on 3.2 GHz. Their dissipated power and area are in the same range, 85 Watt, $253mm^2$ and 30 Watt, 400 mm^2 respectively. **Considering the IBM Cell processor's computing power per watt performance as a unit, computational efficiency of the Xilinx Virtex 5 SX240T FPGA is 22 times higher, while providing 8 times higher performance. The one order of magnitude speedup of the FPGA is owing to the arithmetic units working fully parallel and the number of implementable arithmetic units.** During CFD simulation, the IBM Cell processor and the FPGA based accelerator can achieve 2 and 3 order of magnitude speedup respectively compared to a conventional microprocessor (e.g.: Intel x86 processors).

2. Thesis: *Examination of the precision and the accuracy of*

partial differential equation solver architectures on FPGA: I have shown in my thesis, that significant speedup can be achieved by decreasing the state precision on FPGA. Engineering applications usually does not require 14-15 digit accuracy, therefore the decreased computational precision can be acceptable. Reduction of the state precision makes it possible to map some particularly complex problems onto an FPGA. I have developed a methodology to specify the minimal required computational precision to reach the maximal computing performance on FPGA where the accuracy of the solution and the grid resolution is given a-priori. The required computational precision can only be determined precisely in infrequent cases, when the exact solution is known.

2.1. **I have elaborated a method to find the minimum required computing precision of the arithmetic units when the step size, spacial resolution and the required accuracy is defined. I have given a tested method to find the precision of the arithmetic unit of a problem, which has analytic solution.** For problems without analytic solution, the reduced precision results can be compared to the 64 bit floating point reference precision. The finest resolution of the grid can also be determined by the method if the desired accuracy is defined.

2.2. **I have shown during the solution of the advection equation (1), that higher computing power can**

be achieved at the expense of the precision.

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0 \tag{1}$$

where t denotes time, u is a conserved property, c is the advection speed. **During the investigation of the arithmetic unit of the advection equation solver the precision is decreased from 40 bit to 29 bit, while area requirements of the architecture are decreased by 20-25% independently from the applied discretization method.** Clock frequency of the arithmetic units does not increase significantly due to the decreased precision, the main source of speedup is the increased number of implementable arithmetic units on the FPGA.

2.3. **I have proved experimentally that area requirements of the arithmetic units can be significantly reduced by using properly normalized fixed point numbers. During the investigation of the advection equation solver architecture, error of the solution of the 33 bit fixed point and the 40 bit floating point (29 bit mantissa) arithmetic unit is in the same order, but the area required for the arithmetic unit is decreased by 15 times.** The main source of speedup is the increased number of implementable arithmetic units on the FPGA, when fixed point arithmetic is used.

3. Thesis: *Implementation of a Global Analogical Programming Unit for emulated digital CNN-UM processor on FPGA archi-*

tecture: The dynamics of the CNN can be emulated by the Falcon processor with different computing precision, arbitrary sized template on many layers. It should be extended with Global Analogical Programming Unit (GAPU) in order to execute a more complex analogical algorithm time efficiently, additionally a Vector Processor should be attached to accelerate arithmetic and logic operations. The GAPU is not only used during program organizing and I/O peripheral management tasks but it should execute local logic, arithmetic and analog instructions as well. Furthermore, timing and control signals of the Falcon processor should be set correctly by the GAPU.

The proposed modifications were implemented and verified with a testing example. Due to the implemented modifications and the extension with the GAPU and the Vector Processor, a real image processing system, a Cellular Wave Computer can be developed.

3.1. **I made recommendations for the structure of the GAPU (precision) to develop an emulated digital CNN-UM.** The Falcon processor should be extended with the GAPU, according to the original CNN-UM architecture, in order to execute a more complex algorithm time efficiently. **The implemented GAPU should consume minimal area while providing high operating speed to avoid slow down of the Falcon processor, to gain the largest possible computational performance. The GAPU can be built from a properly configured MicroBlaze, or a dedicated PPC, or ARM processor. I made further considerations on the**

structure of the controller's state registers and configuration of the template and state memory in order to adopt the system for the different kind of Falcon Processing Units. E.g.: Different Falcon units are optimal for black and white or grayscale image processing.

3.2. **I have developed a new architecture, where the embedded microprocessor, the controller circuit, the memory and the Falcon processing unit can be operated on different clock speed. In addition to the internal structural modifications the external memory can be accessed via a dedicated FIFO element. The new architecture makes concurrent access to the external memory possible for the MicroBlaze, the control unit and the Falcon processor.**

The dedicated arithmetic units of the new generation FPGAs become faster, but the speed of the embedded processor and bus architecture are evolving slower. The Falcon processor can work on higher operating frequency than the embedded microprocessor and the bus system on the latest FPGAs.

4.   Application of the results

4.1.   Application of the Fluid Flow Simulation

Simulation of compressible and incompressible fluids is one of the most exciting areas of the solution of PDEs because these equations appear in many important applications in aerodyna-

mics, meteorology, and oceanography. Modeling ocean currents plays a very important role both in medium- term weather forecasting and global climate simulations. In general, ocean models describe the response of the variable density ocean to atmospheric momentum and heat forcing. In the simplest barotropic ocean model a region of the oceans water column is vertically integrated to obtain one value for the vertically different horizontal currents. The more accurate models use several horizontal layers to describe the motion in the deeper regions of the ocean. Such a model is the Princeton Ocean Model (POM), being a sigma coordinate model in which the vertical coordinate is scaled on the water column depth.

Computational Fluid Dynamics (CFD) is the scientific modeling the temporal evolution of gas and fluid flows by exploiting the enormous processing power of computer technology. Simulation of fluid flow over complex shaped objects currently requires several weeks of computing time on high performance supercomputers. The developed CFD simulation architecture, implemented on FPGA, is several order of magnitude faster than todays microprocessors.

4.2. Examining the accuracy of the results

In real life engineering application double precision floating point numbers are used for computations to avoid issues of roundoff error. However it is worth to examine the required precision, if the computing resources, power dissipation or size is limited or the computation should be carried out in real time. The speed of the partial differential equation solver architecture implemented

on FPGA can be greatly increase, if we decrease the precision of the solver architecture, consequently more processing unit can be implemented on the same area. This thesis is useful if we want to investigate the limitation of a real time computation. I have examined a simplified advection equation solver architecture, where the analytic solution is known. With the minimal modification of such problems (which has analytic solution), the computed precision is remaining probably acceptable with a similar problem, which has no analytic solution.

4.3. The importance of Global Analogic Programming Unit

In order to provide high flexibility in CNN computations, it is interesting how we can reach large performance by connecting locally a lot of simple and relatively low-speed parallel processing elements, which are organized in a regular array. The large variety of configurable parameters of this architecture (such as state- and template-precision, size of templates, number of rows and columns of processing elements, number of layers, size of pictures, etc.) allows us to arrange an implementation, which is best suited to the target application (e.g. image/video processing). So far, without the GAPU extension, when solving different types of PDEs, a single set of CNN template operations has been implemented on the host PC: by downloading the image onto the FPGA board (across a quite slow parallel port), computing the transient, and finally uploading the result back to the host computer where logical, arithmetic and program organizing steps were executed.

Reconfigurable CNN-UM implementation on FPGAs may also mean a possible breakthrough point towards industrial applications, due to their simplicity, high computing power, minimal cost, and fast prototyping.

5. Acknowledgements

6. Publications

6.1. The author's journal publications

[1] Z. Nagy, L. Kék, Z. Kincses, A. Kiss, and P. Szolgay, „Toward Exploitation of Cell Multi-processor Array in Time-consuming Applications by Using CNN Model," *International Journal of Circuit Theory and Applications*, vol. 36, no. 5-6, pp. 605–622, 2008.

[2] Z. Vörösházi, A. Kiss, Z. Nagy, and P. Szolgay, „Implementation of Embedded Emulated-Digital CNN-UM Global Analogic Programming Unit on FPGA and its Application," *International Journal of Circuit Theory and Applications*, vol. 36, no. 5-6, pp. 589–603, 2008.

6.2. The author's international conference publications

[3] Z. Vörösházi, Z. Nagy, A. Kiss, and P. Szolgay, „An Embedded CNN-UM Global Analogic Programming Unit Implementation on FPGA," in *Proceedings of the 10th IEEE International Workshop on Cellular Neural Networks and their Applications*, (Istanbul, Turkey), CNNA2006, August 2006.

[4] Z. Vörösházi, A. Kiss, Z. Nagy, and P. Szolgay, „FPGA Based Emulated-Digital CNN-UM Implementation with GAPU," in *Proc. of CNNA'2008*, (Santiago de Compostella), pp. 175–180, 2008.

[5] Z. Nagy, L. Kék, Z. Kincses, A. Kiss, and P. Szolgay, „Toward Exploitation of Cell Multi-Processor Array in Time-Consuming Applications by Using CNN Model," in *Proc. of CNNA'2008*, (Santiago de Compostella), pp. 157–162, 2008.

[6] Z. Vörösházi, A. Kiss, Z. Nagy, and P. Szolgay, „A Standalone FPGA Based Emulated-Digital CNN-UM System," in *Proc. of CNNA'2008*, (Santiago de Compostella), 2008.

[7] Z. Nagy, A. Kiss, S. Kocsárdi, and Á. Csík, „Supersonic Flow Simulation on IBM Cell Processor Based Emulated Digital Cellular Neural Networks," in *Proc. of ISCAS'2009*, (Taipei, Taiwan), pp. 1225–1228, 2009.

[8] Z. Nagy, A. Kiss, S. Kocsárdi, and Á. Csík, „Computational Fluid Flow Simulation on Body Fitted Mesh Geometry with IBM Cell Broadband Engine Architecture," in *Proc. of ECCTD'2009*, (Antalya, Turkey), pp. 827–830, 2009.

[9] Z. Nagy, A. Kiss, S. Kocsárdi, M. Retek, Á. Csík, and P. Szolgay, „A Supersonic Flow Simulation on IBM Cell Processor Based Emulated Digital Cellular Neural Networks," in *Proc. of CMFF'2009*, (Budapest, Hungary), pp. 502–509, 2009.

[10] A. Kiss and Z. Nagy, „Computational Fluid Flow Simulation on Body Fitted Mesh Geometry with FPGA Based Emulated Digital Cellular Neural Networks," in *Proceedings of 12th International Workshop on Cellular Nanoscale Networks and their Applications*, (Berkeley, CA, USA), CNNA2010, 2010.

[11] L. Füredi, Z. Nagy, A. Kiss, and P. Szolgay, „An Improved Emulated Digital CNN Architecture for High Performance

FPGAs," in *Proceedings of the 2010 International Symposium on Nonlinear Theory and its Applications*, (Krakow, Poland), pp. 103–106, NOLTA2010, 2010.

[12] C. Nemes, Z. Nagy, M. Ruszinkó, A. Kiss, and P. Szolgay, „Mapping of High Performance Data-Flow Graphs into Programmable Logic Devices," in *Proceedings of the 2010 International Symposium on Nonlinear Theory and its Applications*, (Krakow, Poland), pp. 99–102, NOLTA2010, 2010.