FPGA: Exploration of the possibilities for the
direct synthesis of concurrent C programs on
high-performance computers in FPGAs

Peter Urbanek and Stefan May

Fachhochschule Nürnberg
Fachbereich efi
Wassertorstr. 10
90489 Nürnberg
peter.urbanek@fh-nuernberg.de

Abstract. This report describes the exploration of the possibilities for the direct
synthesis of concurrent C programs on high-performance computers in FPGAs

Key words: FPGA, VHDL, Handel-C, Nios, OpenMP, “intelligent camera”

1 Introduction

The way of creating suitable prototypes and functional models from application
oriented research results often leads from the design, the simulation and the optimi-
sation on a mainframe to a subsequent conversion in a system consisting of hardware
and software components. In such a case, single chip solutions are prefered, which
integrate the results found on the mainframe in a specific chip.

For this purpose, the software solution running on the mainframe is rewritten
from scratch in a hardware description language such as VHDL. This method is
both time and money consuming. Moreover, during the new implementation, new
events are admitted into the design, which already had been eliminated in the prior
software solution.

It would be an enormous improvement, if the hardware could also be described
in the high-level language C and if concurrent processes could be converted directly.
There are already first approaches on the basis of PCs. One of these is the pro-
gramming language Handel-C [1, 8, 9], developed at Oxford University. It offers the
possibility to synthesize systems described in C into ASICs, especially FPGAs [5].

The ideal conversion procedure would be, if the scientist could solve his prob-
lem, e.g. an algorithm, on the mainframe using its high performance as well as the
excellent development environment and consequently synthesize a chip directly from
these results.
2 Objective

This specific form of downright synthesizing concurrent software, running on a mainframe in silicon, is to be analysed in this paper. We shall start from today’s possibilities which contain plain constrictions. Instead of using mainframe software, we must fall back on C with the extension OpenMP [10], because there are only synthesis tools for C on the hardware side at the moment.

The conclusions drawn from this project could certainly be converted to Fortran, which will only make sense once there are synthesis tools on the hardware side for that language.

Due to its good availability, C in connection with OpenMP was used on the mainframe side and Handel-C was used on the FPGAs of the chip side. The studies are held in general and carry no evaluation of the tools. Only the method of converting mainframe programs is of interest and the effort and quality required for that. Is there an advantage in this new approach over the known redevelopment in VHDL? If this question can be answered positively, then research results which need to be proven by a realisation can be converted faster and safer in the future.

The purpose of this research is to convert concurrent mainframe software, running on several processors, into prototypical hardware designs with FPGAs.

Procedure:

At first the conversion of simple algorithms was examined. Since there are no standards to evaluate the C synthesis on the hardware side, the Handel-C realisations were compared to known processor realisations.

After that an image processing system was built to compare complex algorithms. This image processing system, later appearing under the name “Intelligent Camera”, served as a development platform on the one hand and as a demonstration tool on various fairs and events on the other. It contains all the algorithms which took the intended path from the mainframe to the FPGA. An evaluation rounds off the research.

All project results are covered by examples and proved by measurements. The statements are held in general and can be transfered to other mainframes as well as other FPGAs. Many thoughts can be transfered directly to other algorithmic programming languages.

3 Concurrent Computing on the Mainframe

In comparison to PCs, mainframes differ a lot from each other. For this study however, the differences in architecture are of most importance. Out of the various classifications for rough distinctions, Flynn’s was the most successful because of its simplicity. Due to the high efficiency of standard processors and their swift development, the MIMD principle [4] is the most widely spread over all mainframes today. This group is distinguished between multi-computer systems (distributed memory machines) and multi-processor systems (shared memory machines) [11,12]. A third part consists of computers with virtually shared memory. In this group, computers with distributed address space are united and form a logical address space. With regard to the conversion to FPGAs shared memory systems are easier to handle, although not imperative (see chapter Results).