

Performance and Area Modeling of Complete FPGA Designs in the presence of Loop Transformations*

K.R. Shesha Shayee

Department of Electrical Engineering - Systems Division
University of Southern California, EEB100
Los Angeles, California 90089
sraghuna@usc.edu

Joonseok Park and Pedro C. Diniz

USC Information Sciences Institute
4676 Admiralty Way, Suite 1001
Marina del Rey, California 90292
{joonseok,pedro}@isi.edu

EXTENDED ABSTRACT

Digital image processing algorithms are a good match for direct implementation on FPGAs as current FPGA architectures can naturally match the fine grain parallelism in these applications. Typically, these algorithms are structured as a sequence of operations, expressed in high-level programming languages as tight loop nests. The loops usually define a shifting-window region over which the algorithm applies a simple localized operator (e.g., a differential gradient, or a min/max).

Despite their apparent simplicity, mapping these kernels from a high-level programming language directly to FPGA devices is a challenging task as FPGA internal resources are limited. The application of loop-level transformations, important to expose vast amounts of fine-grain parallelism and data reuse, substantially complicates the complexity of mapping these computations to hardware. These loop transformations interact in non-trivial fashion, exposing different space-time trade-offs. For example loop unrolling, while increasing the opportunities for instruction-level-parallelism also increases the required input bandwidth if functional units are not to be stalled waiting for data.

Exploring the number of available transformations for a given computation in an automated fashion is a very desirable approach. Unfortunately, existing synthesis tools do not have program analyses that are powerful enough to uncover relevant data dependence information required to understand the interaction of several loop transformations when multi-dimensional arrays are involved. These tools do not take into consideration the impact of the transformations on the required input/output bandwidth for complete designs. Lastly, they exhibit prohibitively long Place&Route steps to determine if a given design will fit in the target FPGA device.

* Funded by the National Science Foundation (NSF) under grant number CCR-0209228

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
FCCM'03, April 08-11, 2003, Napa, California, USA.

In this research we focus on the development of fast, yet accurate performance and area modeling of complete FPGA designs that combine analytical, empirical and behavioral estimation techniques. We model the application of a set of important program transformations for image processing algorithms, namely *loop unrolling*, *tiling*, *loop interchanging*, *loop fission* and *array privatization*, and explore pipelined and non-pipelined execution modes. We take into consideration the impact of various transformations, in the presence of limited I/O resources like address generators and external memory data channels, on the performance of a complete design implemented in a FPGA based architecture.

An important aspect of our approach is that the parameters' values involved in all of the modeling can be derived automatically. The parameters' values that depend on the target FPGA device such as the maximum clock rate or the actual implementation of the memory interfaces (e.g., the latency or the number of cycles for a non-pipelined operations) are extracted using existing behavioral synthesis tools. The architecture parameters that define the interaction with the external memory are either predefined (e.g., the read/write operation latency) or determined empirically as is the case of the area of the whole memory interface. The remaining parameters are derived using parallelizing compiler analyses techniques of the source code. In our approach we combined the empirical models for area for the memory interface with area estimates provided by behavioral synthesis thus producing the area estimate for a complete design. We then use our analytical performance models augmented with the execution scheduling information provided by synthesis to predict the performance of the overall implementation.

Preliminary experimental results with a real FPGA-based hardware board reveals that our area and performance modeling correlates very well with our simulation results for the limited set of image processing codes – a binary image correlation, matrix-multiple and image histogramming. Our modeling is capable of identifying performance effects of the interaction of the various loop transformations. These, albeit limited, results suggest the proposed modeling to be a very effective approach that will ultimately allow compilers to quickly explore a wide range of program transformations for selecting feasible and high-performance FPGA designs.