DESIGN OF FAST AND EFFICIENT HYBRID-FPGAs FOR NUMERICALLY INTENSIVE APPLICATIONS IN FLUID DYNAMICS AND IMAGE/VIDEO PROCESSING

A. Akoglu, Aravind R. Dasu, S. Panchanathan
Arizona State University, Ira A. Fulton School of Engineering
Department of Computer Science and Engineering
Center for Cognitive Ubiquitous Computing (CUbiC) Lab
akoglu@asu.edu, dasu@asu.edu, panch@asu.edu

Numerical simulations in Computational Fluid Dynamics, involves modules solving sets of equations (2nd-order Crank-Nicolson/Adams-Bashforth, 3rd-order Runge-Kutta time-stepping etc.) which have some common computation features, and performed iteratively. Similarly video and image processing applications involve tasks (mosaic building to compress video into images, image compression such as DCT, DWT etc.) which also have some common computation features and require iterative processing. Both these domains are known to have a high degree of relevance to military applications. It has been shown by several researchers that these applications are well suited to be executed on spatially parallel processor architectures. FPGAs in particular offer large amounts of on-chip spatial parallel units, thus capable of performing orders of magnitude faster than regular serial processors. But FPGAs suffer from the drawbacks of being application agnostic and hence incur penalties of loss of clock cycles in redundant reconfigurations, generic routing and poor memory architectures which impact speed, power and silicon area. All these factors have led us into exploring the reconfigurable architecture design space with the application domain being prioritized. This paper focuses on the extraction of tasks or core clusters in Control Data Flow Graphs (CDFGs) of multimedia and fluid dynamics applications followed by designing the architecture to embed them in Hybrid-FPGA environments. By Hybrid, we mean that the proposed FPGA architectures will involve LUT regions, ASIC regions and possibly VPGA regions. Tasks or core clusters obtained through the common sub-graph analysis between basic blocks within and across routines are basically recurring computation patterns implemented as ASICs on non-reconfigurable area. After removing the common sub-graphs from the CDFG, remaining parts of each basic block are then implemented on LUT based reconfigurable area. A Packing mechanism designed to improve the routing architecture by reducing the switching requirements by 12-20% has been proposed. This mechanism, for configurable logic uses live-in live-out variable analysis and scheduling information of CDFGs in its cost function in addition to routability and timing driven cost metrics defined by other researchers. We have conducted experiments on several complex routines from the target applications. Map/synthesis reports based on Xilinx architectures were obtained. Results show that partial reconfiguration with the use of computation cores embedded in a sea of LUTs offer the potential for massive savings in gate density. In addition to that, by eliminating the need for unnecessary and redundant sub-circuit pattern configurations, switching requirements in configurable area is reduced due to localization of global connections.

Designing processing elements based on identifying correlated compute intensive regions within each application and between applications result in large amounts of processing in localized regions of the chip. This reduces the amount of reconfigurations and on-chip communication hence results with faster application switching and reduced power consumption. This task comprises of finding the Largest Common Sub-graph, which is a NP complete problem. Proposed methods on node growing [1,2] for graph searches in databases or template matching based clustering [3, 4, 5 and 6] attempt to solve this problem. Core reusable regions that have been detected as common within or across applications by peer research efforts, have either been at the granularity of MAC units (2 nodes) or at the granularity of entire function modules. There has been no reported work that has detected core reusable regions consisting of several operation (multiply, add, divide etc) nodes between basic blocks in applications. Our method generates ASIC cores of higher granularity by specifically focusing on Dataflow graphs of Hardware Computations and taking advantages of the restrictions that they offer. In Hybrid-FPGA model, we propose that the CIPE region be constrained and mapped onto a slab (a region of LUTs isolated by MacroBus as in the Virtex architecture) or implemented as gates in ASIC technology. Even though a large ASIC on chip increases the costs of mask design, it offers the maximum amount of gate savings. Remaining slabs are implemented on LUT based reconfigurable. To the best of our knowledge currently there exists no known technology that maps regions within a single DFG (corresponding to a BB) into multiple Slabs for Partial Reconfiguration.

In this paper we also address the issue of packing mechanism for LUTs on reconfigurable area. For fine-grained logic, more logic blocks will be required to implement the circuit. Routing area may become excessive. In coarse-grained logic, much of the logic functionality may be unused wasting area. A cost function is needed to make the decision of inserting the building block into one of the candidate clusters. In addition to having variable size building blocks, we take into account the control data flow graph of each possible execution path and do a live-in live-out variable analysis between basic blocks [7]. To our knowledge no work has been done in this direction. Cost function simplifies the complexity of the placement and routing steps since constraints of these steps are evaluated as

early as at the packing step. Several time or area driven packing with bottom-up or top-down approaches have been proposed earlier by other researchers [8, 9, 10 and 11]. Those approaches incorporate connectivity, routability metric such as density of high fan out nets, traffic in and out of the logic block, number of nets and connectivity into packing cost function. We propose that cost of adding a building block into a cluster depends on not only the metrics listed above but also how timing of the circuit is affected at different possible execution paths (the scheduling information), data dependency between basic blocks in reconfigurable region between CPIE modules, and between CPIE & non-CPIE modules.

References

[1] M. Kuramochi, G. Karypis, Technical Report 02-026, Department of Computer Science/Army HPC Research Center, University of Minnesota, June 27, 2002.

[2] R. Kastner et. al, "Instruction Generation for Hybrid Reconfigurable Systems,", ICCAD 2001.

[3] K. Keutzer, "Technology Binding and Local Optimization by DAG Matching" Proc. of DAC87

[4] A. Chowdhary, S. Kale, P. Saripella, N. Sehgal and R. Gupta, "A General Approach for Regularity Extraction in Datapath Circuits," Proc. Of ICCAD, 1998.

[5] D. S. Rao and F. J. Kurdahi. "On Clustering for Maximal Regularity Extraction," IEEE Trans. on Computer-Aided Design, Vol. 12, No. 8, August, 1993.

[6] S. Cadambi and S. C. Goldstein. "CPR: A Configuration Profiling Tool," Proc. of the Symposium on Field-Programmable Custom Computing Machines, 1999.

[7] A. Akoglu, A. Dasu and S. Panchanathan, "A Framework for the Design of the Heterogeneous Hierarchical Routing Architecture of a Dynamically Reconfigurable Application Specific Media Processor", Workshop on Embedded Systems for Media Processing International Conference on High Performance Computing (HiPC), Dec 17, 2003,Hyderabad, India

[8] V. Betz and J. Rose, "VPR: A New Packing Placement and routing Tool for FPGA Research", International Workshop on Field-Programmable Logic and Application, pp. 213-222, 1997.

[9] A. Marquardt , V. Betz and J. Rose, "Using cluster-based logic blocks and timing-driven packing to improve FPGA speed and density", Proceedings of the 1999 ACM/SIGDA seventh international symposium on Field programmable gate arrays, p.37-46, February 21-23, 1999, Monterey

[10] E. Bozorgzadeh, S. Ogrenci-Memik and M.Sarrafzadeh, "RPack: routability-driven packing for cluster-based FPGAs", Proceedings of the conference on Asia South Pacific Design Automation Conference, p.629-634, January 2001, Japan.

[11] A. Singh, G. Parthasarathy and M. Marek-Sadowska, **"**Efficient circuit clustering for area and power reduction in FPGAs" ACM Transactions on Design Automation of Electronic Systems**,** Vol 7, I: 4 ,October 2002, pp: 643 – 663.