A NOVEL HEURISTIC AND PROVABLE BOUNDS FOR RECONFIGURABLE ARCHITECTURE DESIGN

Alastair M. Smith, George A. Constantinides, and Peter Y. K. Cheung

Department of Electrical and Electronic Engineering,
Imperial College London
email: {alastair.smith, g.constantinides, p.cheung} @imperial.ac.uk

ABSTRACT

This paper is concerned with the application of formal optimisation methods to the design of mixed-granularity FPGAs. In particular, we investigate the appropriate mix and floorplan of heterogeneous elements: multipliers, RAMs, and LUT-based logic, in order to maximise the performance of a set of DSP benchmark applications, given a fixed silicon budget. We extend our previous mathematical programming framework by proposing a novel set of heuristics, capable of providing upper-bounds on the achievable reconfigurable-to-fixed-logic performance ratio. Our results provide, for the first time, quantifications of the optimal performance/area-enhancing capability of multipliers and RAM blocks within a system context, and indicate that only a minimal performance benefit can be achieved over Virtex II by re-organising the device floorplan, when using optimal technology mapping.

1. INTRODUCTION

Field-programmable gate arrays (FPGAs) are commonly used for high throughput computation. Traditionally, FPGAs have consisted of Look-Up Tables (LUTs) capable of performing any four input logic function. Recent introductions into the FPGA fabric, such as DSP blocks and RAM, have been used to speed up computation or take advantage of greater logic density. There has been considerable research into exploring the architectures of homogeneous LUT-based FPGA devices. This has concentrated on exploring the nature of the LUTs, for instance how many inputs they use, and how they are locally interconnected. In this paper, the emphasis is on exploring architectures by examining the ratios and physical placements of the different components that are found in heterogeneous devices.

When designing a new device architecture, it is common for the architects to have a base-line parameterisable structure from which many different architectures can be generated. A variety of possible architectures are then simulated, with reference designs placed and routed in each architecture using heuristics such as simulated annealing. The final architecture will be one from this set that best suits the area, speed and power consumption metrics for all designs. By using linear programming (LP) and integer linear programming (ILP), it is possible to simultaneously place benchmarks and generate heterogeneous architectures, *as well as* perform module selection for given computational structures in a benchmark; *e.g.* decide whether a ROM should be implemented in LUTs or in an embedded component [2]. The combined approach eliminates both the need for exhaustive testing on a set of architectures and the dependence on heuristic parameters.

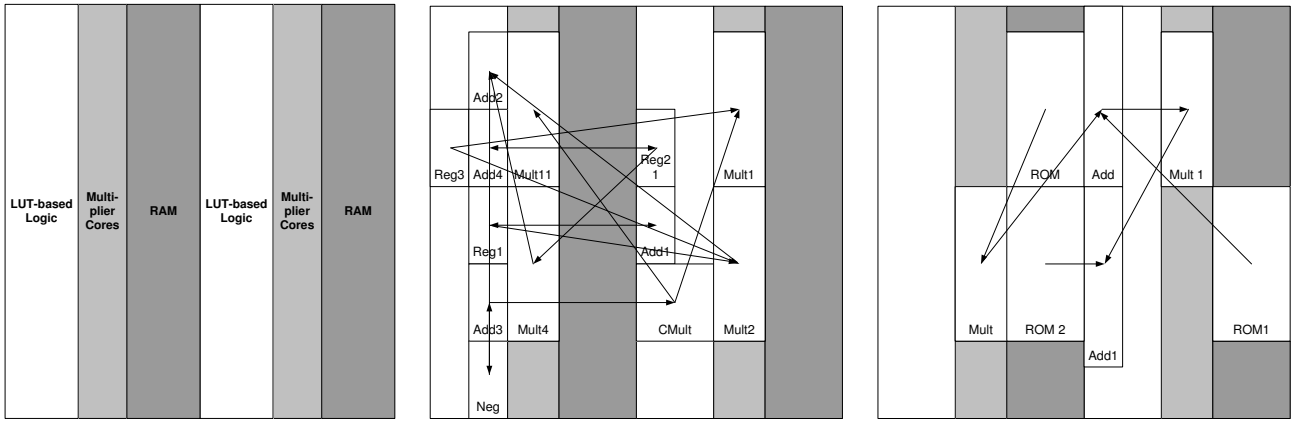
This work has been funded by the EPSRC (UK) under grant number EP/C549481/1 and under the EPSRC DTA scheme.

2. LINEAR PROGRAMMING AND ARCHITECTURE DEVELOPMENT

The main objective of the architecture development system is to minimise the clock period of designs. The objective function of the optimisation is a measure of clock period that can be related across a benchmark set, and is relative to the best achievable with reconfigurable components. Circuit delays of the computational structures in circuits (nodes) can be modelled in linear programming by using Bellman's equations [1]. Bellman's equations can be formulated in a way so as to incorporate module selection [2].

Similarly, floorplanning issues for heterogeneous FPGA architectures have been modelled within a linear programming environment [2]. The floorplanning constraints within the ILP can be used to determine both the optimal architecture floorplan and the floorplan of each benchmark for which the architecture is to be designed for. An illustrative example of the problem is given in Figure 1, in which an architecture has been generated using the proposed linear programming framework. The figure also shows the floorplans of the two benchmarks that the architecture was generated for.

The combination of the linear constraints allow benchmark floorplanning, module selection and architecture generation to be performed concurrently. In addition, it is possible to include constraints to model particular architectures.



(a) (b) (c)

Fig. 1. A simple scaled-down example floorplan of a throughput-optimal architecture/mapping combination generated by our work for a particular area constraint. (a) The architecture, (b) A mapping of a 1st order LMS adaptive filter, (c) A mapping for a 2nd order polynomial evaluation. Arrows indicate the data-flowdependencies.

For instance, the architecture can be specified to have no multiplier regions, in which case the automated module selection will expand any multipliers in the benchmarks into LUT-based implementations. Commercial architectures can be modelled in a similar manner by specifying the locations and widths of the regions of each resource type.

3. HEURISTIC DETERMINATION OF RECONFIGURABLE ARCHITECTURES

Recent work on this project has been concerned with achieving scalable run-time. The run-time of the ILP solver makes a direct solution of the ILP unattractive for large benchmark sets ($\gg$ 24 hours). As a consequence, it was necessary to develop a methodology to counter this problem. The ILP framework allows the development of a heuristic approach in a structured manner.

In ILP, it is the integer variables that have the most significant affect on run-time. The heuristic approach to solution of the ILP is based on relaxation of integer variables related to benchmark floorplanning to real values, allowing solution through, for example, the Simplex method [3]. The heuristic is used to iteratively select which of these relaxed integer variables to round. The iterative nature of this procedure allows a gradual crystallisation of the architecture.

Another aspect of the heuristic, introduced in order to guarantee scalable run-time, is a clustering phase. This reduces run-time by removing the integer variables associated with architecture floorplanning. The architectures generated are those in which the heterogeneous resources are grouped into columns, as illustrated in Figure 1(a). The clustering heuristic assigns the locations of the columns of the different resource types, and also assigns the computational structures of the benchmarks into the different regions. The widths of these regions are determined by the ILP solver.

4. RESULTS

Using the architecture development system, it has been possible to perform several interesting experiments. To quantify the advantages that embedded multipliers provide, the proposed system was used to map to architectures without embedded multipliers. This showed that multipliers can reduce the clock period by as much as 60%, with area savings of up to 680% times. The proposed system was also used to show that embedded memory can significantly improve the area efficiency of reconfigurable architectures, although the clock period improvements are significantly less, in this case around 5%. Finally, the proposed system has been used to model members of the Virtex 2 family. This has shown that the Virtex2 XC2V2000 is within 2.5% of the optimal given the set of benchmarks and the set components on the Virtex 2 device. Further improvements can only be made by modifying the existing components, or incorporating new components within the fabric.

5. REFERENCES

[1] G. De Micheli, *Synthesis and Optimization of Digital Circuits*, McGraw-Hill, 1994.

[2] A. M. Smith, G. A. Constantinides, and P. Y. K. Cheung, "Generation and exploration of reconfigurable architectures using mathematical programming," in *Field-Programmable Logic and Applications*, 2005.

[3] R. S. Garfinkel and G. L. Nemhauser, *Integer Programming*, John Wiley and Sons, Inc., New York, 1972.