



HAL
open science

An automated design approach to map applications on CGRAs

Thomas Peyret, Gwenolé Corre, Mathieu Thevenin, Kevin Martin, Philippe
Coussy

► **To cite this version:**

Thomas Peyret, Gwenolé Corre, Mathieu Thevenin, Kevin Martin, Philippe Coussy. An automated design approach to map applications on CGRAs. GLSVLSI Great Lakes Symposium on VLSI, May 2014, Houston, Texas, United States. pp.229-230, 10.1145/2591513.2591552 . hal-01002316

HAL Id: hal-01002316

<https://hal.science/hal-01002316v1>

Submitted on 5 Jun 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An Automated Design Approach to Map Applications on CGRAs

Thomas Peyret, Gwenolé Corre, Mathieu Thevenin
CEA, LIST, Electronic Architectures and Sensors Laboratory
F-91191 Gif-sur-Yvette, France
+33 (0)1 69 08 27 85
firstname.lastname@cea.fr

Kevin Martin, Philippe Coussy
Université de Bretagne-Sud, Lab-STICC
Lorient, France
+33 (0)2 97 87 45 65
firstname.lastname@univ-ubs.fr

ABSTRACT

Coarse-Grained Reconfigurable Architectures (CGRAs) are promising high-performance and power-efficient platforms. However, their uses are still limited by the capability of mapping tools. This abstract paper outlines a new automated design flow to map applications on CGRAs. The interest of our method is shown through comparison with state of the art approaches.

Categories and Subject Descriptors

B.5.2 [Register-Transfer-Level Implementation]: Design Aids – Automatic synthesis.

Keywords

CGRA; Mapping; Scheduling; Binding;

1. INTRODUCTION

For the last two decades, Coarse-Grained Reconfigurable Architectures (CGRAs) have been mainly proposed for accelerating multimedia applications. CGRA are indeed an interesting trade-off between FPGAs and many-core architectures thanks to their power efficiency and programmability [9]. The literature is very rich in CGRAs architectures, which distinguish by different features such as the granularity of the Processing Elements (PE) named *tile*, homogeneity or heterogeneity of PE, type of operators, absence/presence of Register Files (RF) or interconnection network topologies. Figure 1 presents an example of CGRA.

The result of the “compilation” of an application on a CGRA (named *mapping*) is the scheduling and the binding of its operations on operators and registers. This NP-complete process [4] must be automated to allow efficient mapping of complex applications. Several methods have been proposed to tackle this problem. They are split in two categories i.e. (1) approaches that solve scheduling and binding separately with heuristics or meta-heuristics [2, 4, 7] or by combining an heuristic and an exact method [3] and (2) approaches that solve the whole problem entirely with exact method [1] or meta-heuristics [6, 8].

Permission to make digital or hard copies of part or all 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. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s). Copyright is held by the author/owner(s).
GLSVLSI'14, May 21–23, 2014, Houston, Texas, USA.
ACM 978-1-4503-2816-6/14/05.
<http://dx.doi.org/10.1145/2591513.2591552>

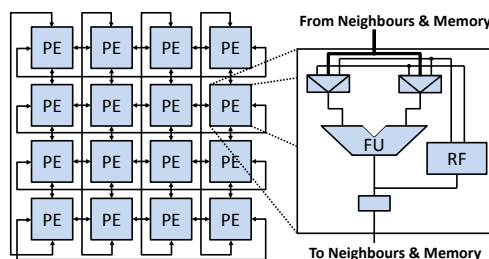


Figure 1 A 4×4 CGRA with 2D mesh torus and RF in each tile

This paper presents a unified approach that maps application on CGRAs. The proposed mapping flow relies on simultaneous scheduling and binding steps respectively based on a heuristic and an exact method followed by a pruning step. The graph of the application is backward traversed and dynamically transformed allowing a better exploration of the design space. This extended abstract paper is organized as follows. Section 2 depicts proposed method. Section 3 presents the experiments and discusses obtained results. Conclusion is given in Section 4.

2. PROPOSED METHOD

Our design flow is presented in Figure 2. Inputs are a C/C++ application code compiled to obtain a formal Control Data Flow Graph (CDFG) and the targeted CGRA’s model. Objective of the method is to minimize latency under resource constraint. The proposed mapping approach allows exploring the design space while keeping computation time low.

The key idea is to combine the advantages of exact and heuristic methods while minimizing as much as possible their respective drawbacks. CDFG is mapped by processing each Data Flow Graph (DFG) of basic bloc sequentially. A list-scheduling based algorithm schedules nodes of each DFG. As it is a local greedy method, the binding is made simultaneously to ensure that at least one solution exists, hence avoiding dead-ends, and is realized incrementally by using an exact method derived from Levi’s algorithm [5]. However, as exact methods do not scale up [4], a wise pruning step is executed at the end of each scheduling cycle to remove redundant partial mappings and thus keep a reasonable number of solutions during mapping process. Besides, DFGs are dynamically transformed as needed when no mapping (*i.e.* during scheduling or binding) solution is found. DFGs are also backward traversed to allow for using more different graph transformations.

3. EXPERIMENTS AND RESULTS

The proposed synthesis flow has been fully automated using Java. GCC has been used to generate CDFGs from applications. Five applications from signal processing domain have been used for our experiments: DC filter, Elliptic filter, Moving Exponential

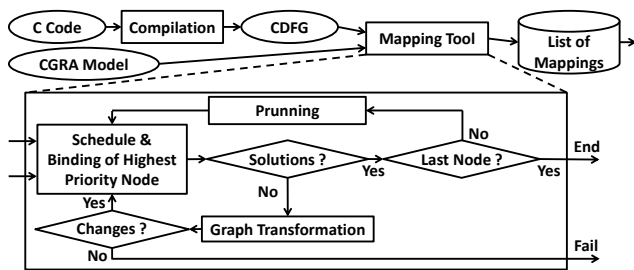


Figure 4 General Flow and Algorithm Core

Average Filter (EMA), Moving Window Deconvolution (MWD) and unsharp mask. To obtain a large spectrum of results, several constraints have been considered: CGRA size, RF size and the number of tiles the final mapping is allowed to use leading to 16 different set of constraints per application and per method.

The proposed approach is compared with two approaches from state of the art. The first, named “Method 1”, solves the scheduling and the binding problem separately as the initial step of [4]. It uses a forward list scheduling algorithm and binding is made by using Levi’s algorithm. “Method 2” forward traverses the graph, schedules nodes by applying statically graph transformations and tries to find a mapping by using Levi’s algorithm as proposed in [3] (that have been shown to provide better results than [8]).

Two metrics were considered: (1) success rate (percentage of time the method finds a solution when at least one of the compared methods succeeds) and (2) percentage of time the method gives the best latency between the compared methods. Figure 3 and Figure 4 give the comparisons between the three methods for the previously defined metrics.

Figure 3 shows that Method 1, which solves scheduling and binding totally separately, leads to the lowest success rate (~56%). Method 2, which transforms the graph *a priori*, provides better results (~67%) but is not as good as the proposed approach (~98%). Figure 4 shows the percentage of time each

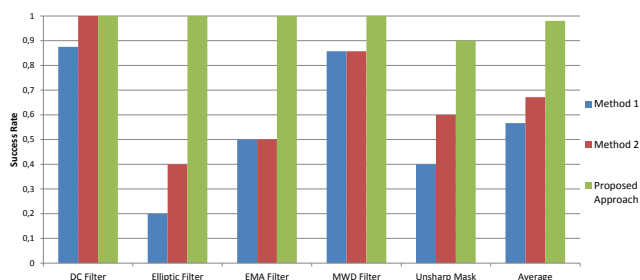


Figure 2 Success Rate

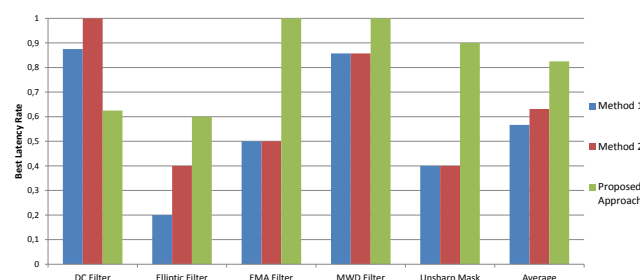


Figure 3 Percentage of a method for obtaining the best latency

method found the best latency and shows that the Proposed Method finds it most of the time (~82%) even if it relies on a heuristic-based scheduling algorithm, while the Methods 1 and 2 find it for respectively 57% and 63% of the benchmark.

4. CONCLUSION

In this paper, a generic method to map applications written in high level language on CGRA architectures has been presented. Experimental results show that this method finds 82% of time the best latency, has the highest success rate and achieves 2.2 times better mappings throughput compared to the other methods and thus achieves a very good exploration of the solution space.

5. REFERENCES

- [1] Brenner, J.A., Veen, J.C. van der, Fekete, S.P., Oliveira Filho, J. and Rosenstiel, W. 2006. Optimal Simultaneous Scheduling, Binding and Routing for Processor-like Reconfigurable Architectures. *Field Programmable Logic and Applications, International Conference on* (2006).
- [2] Friedman, S., Carroll, A., Van Essen, B., Ebeling, C., Hauck, S. and Ylvisaker, B. 2009. SPR: an architecture-adaptive CGRA mapping tool. *Proceedings of the ACM/SIGDA international symposium on Field programmable gate arrays* (2009), 191–200.
- [3] Hamzeh, M., Shrivastava, A. and Vrudhula, S. 2013. REGIMap: register-aware application mapping on coarse-grained reconfigurable architectures (CGRAs). *Design Automation Conference* (2013).
- [4] Lee, G., Choi, K. and Dutt, N.D. 2011. Mapping multi-domain applications onto coarse-grained reconfigurable architectures. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*. 30, 5 (2011), 637–650.
- [5] Levi, G. 1973. A note on the derivation of maximal common subgraphs of two directed or undirected graphs. *Calcolo*. 9, 4 (Dec. 1973), 341–352.
- [6] Mei, B., Vernalde, S., Verkest, D., De Man, H. and Lauwereins, R. 2002. DRESC: A retargetable compiler for coarse-grained reconfigurable architectures. *Field-Programmable Technology, 2002. (FPT). IEEE International Conference on* (2002), 166–173.
- [7] Park, H., Fan, K., Mahlke, S.A., Oh, T., Kim, H. and Kim, H.-S. 2008. Edge-centric modulo scheduling for coarse-grained reconfigurable architectures. *Proceedings of the 17th international conference on Parallel architectures and compilation techniques* (2008).
- [8] De Sutter, B., Coene, P., Vander Aa, T. and Mei, B. 2008. Placement-and-routing-based register allocation for coarse-grained reconfigurable arrays. *ACM SIGPLAN Notices*. 43, 7 (Jun. 2008), 151.
- [9] Taylor, M.B. 2012. Is dark silicon useful?: harnessing the four horsemen of the coming dark silicon apocalypse. *Design Automation Conference* (2012).