High-Level Synthesis Methodologies for Delay-Area Optimized Coarse-Grained Reconfigurable Coprocessor Architectures

Sotirios Xydis (PhD Candidate), Kiamal Pekmestzi (Supervisor), Dimitrios Soudris (Co-advisor), George Economakos (Co-advisor)
Microprocessors and Digital Systems Laboratory, Electrical and Computer Engineering Department, National Technical University of Athens, Zographou Campus, Athens Greece 15780, Tel: +30-2107723653, (email: {sxydis, pekmes, dsoudris, geconom}@microlab.ntua.gr)

This PhD thesis was partially supported by the E.C funded MOSART IST-215244 project, Website: http://www.mosart-project.org/

I. Extended Abstract

As Very Large Scale Integration (VLSI) process technology continues to scale down transistor sizes, modern computing devices are becoming extremely complex. In order to face this complexity explosion, the shifting of design methodologies towards higher level of abstraction has been proposed. This high level view of the design procedure enables the automated synthesis of applications' architecture that is written in an application-level description i.e. C/C++. Additionally, it allows designers to explore the tradeoffs between different system and implementation parameters to conclude in an efficient design solution.

The work done during this PhD thesis targets the exploration and optimization of the design solutions in a global manner, by focusing on the combined development of novel ***(i) system-level automated design methodologies/tools and (ii) circuit-level techniques for a specific class of system architectures - reconfigurable systems.***

Reconfigurable Computing has been proposed as a new paradigm to address the conflicting design requirements for high performance and area efficiency. Towards this direction, fine- and coarse-grained reconfigurable coprocessor architectures have been presented [1]. Unlike fine-grained, coarse-grained architectures (CGA) operate at the word level of granularity exhibiting better power and performance features, close to ASIC solutions [1].

However, a performance-area-power gap still exists for CGAs to overcome ASIC implementations [2]. Thus, new fundamental design problems/questions has been raised. *Does this gap be a bridgeable one? How can CGAs shift even closer to ASIC datapaths?* In order to address the aforementioned problems, we identified that hardware sharing at the bit-level generates CGAs with performance and area characteristics closer to ASICs than the existing ones. Thus, this thesis proposes new architectural templates and the corresponding high level synthesis methodologies to enable a new shifting on the state-of-the-art of CGAs.

II. Problem Description

Section I discussed our basic research field of interest. In this Section, we present the specific problems addressed by the proposed PhD thesis.

Problem 1. *Existing CGAs take into account only sharing at the Functional Unit (FU) level between identical types of FUs, thus delivering high area overheads.*

Solution. A novel circuit level design technique, called Flexibility Inlining, has been developed which generates area efficient CGAs templates taking into consideration bit-level hardware sharing [3]. Flexibility Inlining enables both (i) the classical sharing between identical type of FUs together with (ii) the hardware sharing between of FUs of different type and functionality (i.e. adders and multipliers), based on the efficient datapath merging of single or chained Carry-Save (CS) additions/subtractions and CS array-multiplication. Common interconnection schemes among the merged arithmetic operators are exposed through a series of architectural- and bit-level transformations, which exploits the structural symmetries. Thus, operation sharing is enabled with zero overhead on the interconnection scheme of the original arithmetic datapaths.

The two-level hardware sharing enabled by Flexibility Inlining results in very area efficient CGAs without compromising execution delay of the reconfigurable datapath (Fig. 1).

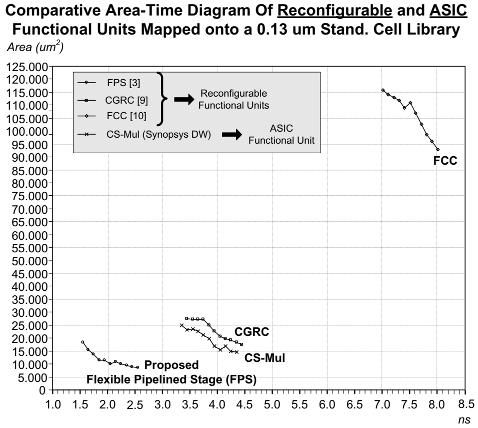


Fig. 1. Area-Performance diagram.

Problem 2. *Existing CGAs do not take into account arithmetic level optimizations during architecture specification leading to major delay overheads in comparison with ASICs.*

Solution. The Carry-Save (CS) arithmetic optimization technique has been considered during architectural design of CGAs. CS arithmetic optimization has been enabled at the bit-level sharing during the appliance of Flexibility Inlining technique [3]. In addition, existing CGA design methodologies based on single FU sharing have been extended in order to take into account CS arithmetic optimization [4].

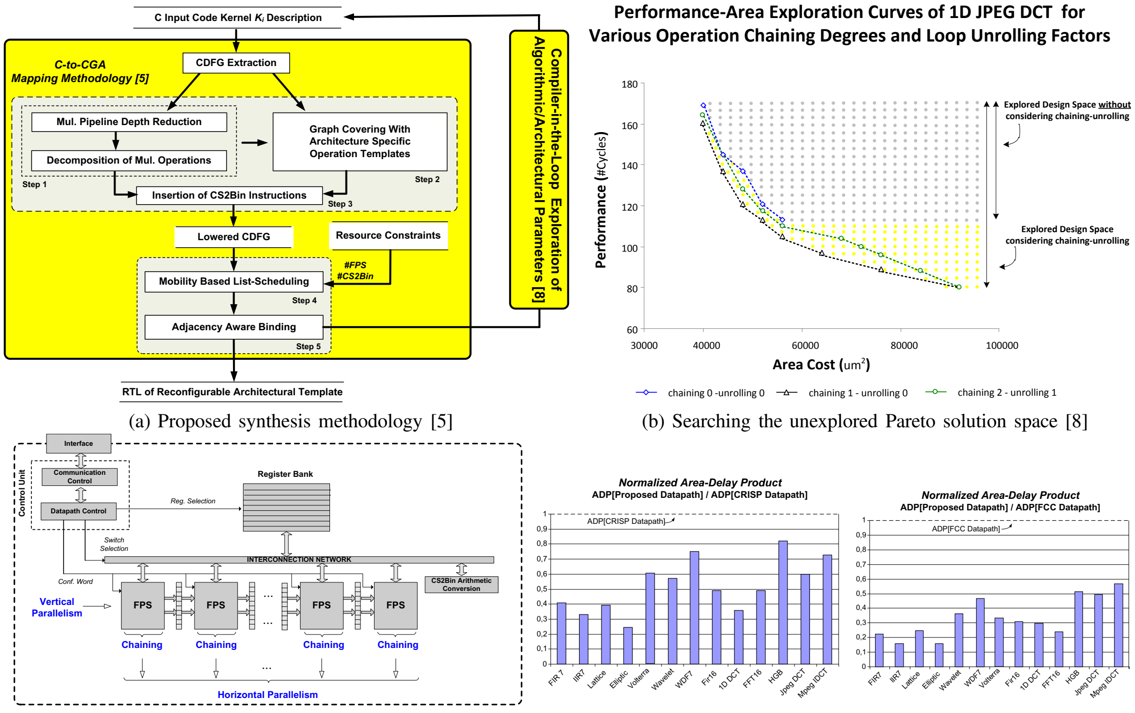
The CS arithmetic level optimization results in high speed reconfigurable units equal or even more efficient than its ASIC equivalents in terms of operating frequency (Fig. 1).

The techniques developed for Problem 1 and Problem 2 give solution to a fundamental problem in circuit design: How to design area efficient circuits without compromising operating frequency and vice versa?

Problem 3. *Productivity and strict time-to-market requirements impose severe constraints in the adoption of new architectural templates based on complex resource models. Thus, optimized solutions (i.e. CGA) are often excluded due to their programming complexity imposed by the underlying hardware resource model.*

Solution. New high-level synthesis algorithms and CAD tools have been developed [5], [6] which automatically map behavioral descriptions (i.e. C/C++) onto optimized CGA architectural templates. The synthesis algorithms take into consideration the inherent features of the CGA template delivering optimized mappings for each application kernel (Fig. 2a). A new design strategy for realizing flexible datapaths has been introduced [5]. For the first time, the combined exploitation of the architectural optimizations: (i) horizontal parallelism, (ii) vertical parallelism and (iii) operation chaining has been considered (Fig. 2c). *Consequently, we filled an existing gap found in current datapath synthesis literature, since the aforementioned architectural optimizations did not treated in a holistic way.* In case of multiple (mutual exclusive) application kernels, datapath merging [7] is performed to efficiently share the datapath (FPS units) and the storage (Register Bank) resources of the CGA coprocessor (Fig. 2c).

978-0-7695-4076-4/10 $26.00 © 2010 IEEE
DOI 10.1109/ISVLSI.2010.8


C Input Code Kernel K_i Description
C-to-CGA Mapping Methodology [5]
CDFG Extraction
Mul. Pipeline Depth Reduction
Decomposition of Mul. Operations
Graph Covering With Architecture Specific Operation Templates
Step 1
Step 2
Insertion of CS2Bin Instructions
Step 3
Lowered CDFG
Resource Constraints
#FPS
#CS2Bin
Mobility Based List-Scheduling
Step 4
Adjacency Aware Binding
Step 5
RTL of Reconfigurable Architectural Template
Compiler-in-the-Loop Exploration of Algorithmic/Architectural Parameters [8]
Performance-Area Exploration Curves of 1D JPEG DCT for Various Operation Chaining Degrees and Loop Unrolling Factors
Performance (#Cycles)
Area Cost (um²)
Explored Design Space without considering chaining-unrolling
Explored Design Space considering chaining-unrolling
chaining 0 -unrolling 0
chaining 1 - unrolling 0
chaining 2 - unrolling 1
Interface
Control Unit
Communication Control
Datapath Control
Register Bank
Reg. Selection
Switch Selection
INTERCONNECTION NETWORK
Conf. Word
CS2Bin Arithmetic Conversion
Vertical Parallelism
FPS
Chaining
Horizontal Parallelism
Normalized Area-Delay Product
ADP[Proposed Datapath] / ADP[CRISP Datapath]
ADP[CRISP Datapath]
FIR 7
IIR7
Lattice
Elliptic
Volterra
Wavelet
WDF7
Fir16
1D DCT
FFT16
HGB
Jpeg DCT
Mpeg IDCT
Normalized Area-Delay Product
ADP[Proposed Datapath] / ADP[FCC Datapath]
ADP[FCC Datapath]

(a) Proposed synthesis methodology [5]

(b) Searching the unexplored Pareto solution space [8]

(c) Flexible coprocessor architecture

(d) ADP: Proposed vs CRISP [9]

(e) ADP: Proposed vs FCC [10]

The complexity of the underlying CGA's hardware resource model is becoming transparent to the designer. By this way, large applications can be coded in standard programming languages and then mapped onto optimized CGA solutions no matter how complex is their resource model. Thus, higher design productivity is achieved for reconfigurable datapaths.

Furthermore, the proposed methodology delivers high quality coprocessor solutions [3], [5], *achieving significant gains in both execution time and area complexity compared to state-of-the-art CGAs i.e. [9], [10]*. Specifically, comparative results shown that in average the proposed CGA mapping and datapath solution delivers in average 44% latency gains together with 37% area reductions [3], [5]. Figs. 2d, 2e depict the shifting towards higher quality design solutions delivered by the proposed methodologies through a normalized comparison with the state-of-the-art CGA coprocessors.

Problem 4. *Existing system level synthesis approaches do not take into account trade-offs produced by the combined impact of architecture level optimizations together with behavioral level optimizations. This results to a large unexplored fraction of the design space and to the loss of optimal solutions.*

Solution. A new augmented design space has been modeled considering both architecture and behavioral level optimizations [8]. In order to efficiently traverse the augmented design space, efficient exploration techniques have been developed, which in each case deliver more optimized design solutions than the existing exploration approaches and converge to the global optima in a quick manner (Fig. 2b). Appropriate CAD tools have been developed automating the proposed design space exploration methodology. The overall exploration procedure is transparent to designer and thus the discovery and convergence towards higher quality solutions comes along with high productivity gains.

III. Conclusions and Future Directions

This PhD thesis targets the design space of automated architectural synthesis for coarse-grained reconfigurable coprocessors. We revaluate the whole design directives proposed in literature [1], proposing arithmetically optimized CGAs based on a combined functional unit and bit-level sharing. This novel view highly differentiates our work from conventional CGA coprocessor approaches which consider only the functional unit sharing.

Future directions of this work will focus on exploiting the proposed architectures and tools to target new design challenges i.e. thermal-awareness, process variability compensation etc. Design automation tools form the appropriate candidates to analyze physical information and make decisions during design time, while CGAs form excellent candidates to dynamically adapt their structure to optimal design points during various run-time situations.

References

[1] S. Vassiliadis and D. Soudris, "Fine- and Coarse-Grain Reconfigurable Computing," *Springer*, pp. 3–149, 2007.

[2] F. Catthoor et al., "IMEC ILP Technology ADRES - DRESC," *Prof. Catthoor talks 2008, available at: www.microlab.ntua.gr/files/imec-catthoor-talk-may-2008.ppt*.

[3] S. Xydis, G. Economakos, and K. Pekmestzi, "Designing Coarse-Grain Reconfigurable Architectures by Inlining Flexibility into Custom Arithmetic Data-paths," *Integration, the VLSI Journal*, vol. 42, no. 4, pp. 486–503, 2009.

[4] S. Xydis, I. Triantafyllou, G. Economakos, K. Pekmestzi, "Flexible Datapath Synthesis Through Arithmetically Optimized Operation Chaining," in *Proc. IEEE NASA/ESA International Conference on Adaptive Hardware and Systems (AHS 2009)*, 2009, pp. 407–414.

[5] S. Xydis, G. Economakos, D. Soudris, and K. Pekmestzi, "High Perforamnce and Area Efficient Flexible DSP Datapath Synthesis," *accepted manuscript to IEEE Trans. on VLSI, available at: www.microlab.ntua.gr/ sxydis/xydisTVLSI09.pdf*.

[6] G. Economakos, S. Xydis, I. Koutras and D. Soudris, "Construction of Dual Mode Components for Reconfiguration Aware High-Level Synthesis," in *Proc. ACM/IEEE Design Automation Conference (DATE'10)*, 2010.

[7] N. Moreano, E. Borin, C. de Souza, and G. Araujo, "Efficient datapath merging for partially reconfigurable architectures," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 24, no. 7, pp. 969–980, 2005.

[8] S. Xydis, C. Skouroumounis, Kiamal Pekmestzi, Dimitrios Soudris, George Economakos, "Designing Efficient DSP Datapaths Through Compiler-in-the-Loop Exploration Methodology," in *to be presented in Proc. of IEEE ISCAS'10*, 2010.

[9] F. Barat, M. Jayapala, T. Vander, R. Lauwereins, G. Deconinck, and H. Corporaal, "Low Power Coarse-Grained Reconfigurable Instruction Set Processor," in *Proc. of FPL'03*. Springer, 2003, pp. 230–239.

[10] M. Galanis, G. Theodoridis, S. Tragoudas, and C. Goutis, "A High Performance Data-Path for Synthesizing DSP Kernels," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 6, pp. 1154–1163, Jun. 2006.