Final report for Ceniit project 01.06 (2001–2007)

Optimization Algorithms for Integrated Code Generation

Christoph Kessler
PELAB, IDA, University of Linköping
e-mail: chrke@ida.liu.se

January 25, 2008

In this project we developed integrative methods for generating high-quality code for embedded and instruction-level parallel processor architectures, including regular architectures such as superscalars and VLIW, as well as irregular architectures, such as clustered VLIW and DSP processors.

Even though today's DSP processors are more compiler friendly than a decade ago or than today's network processors, they impose intricate structural constraints on the parallel use of functional units, the placement of operands in registers, and the scheduling of instructions. Additionally, these various constraints strongly depend on each other. As a consequence, traditional approaches to code generation that solve the problem by treating the subproblems in isolation in separate, subsequent phases, can either not be applied at all or produce code of very poor quality. On the other hand, DSP code must meet very high quality demands in terms of execution time, required memory space, code size, and energy consumption. Often, the application code is fixed, its time-critical part is comparably small in size, and aggressive overnight or parallel optimization is feasible. However, an integrated approach to optimal code generation that considers instruction selection, instruction scheduling and register allocation simultaneously constitutes a very hard combinatorial optimization problem. We developed and implemented exact algorithms, based on dynamic programming and on integer linear programming, that can solve the integrated problem for reasonable problem sizes and realistic architectures. For larger problem instances we provided heuristic algorithms. Beyond execution time, we also targeted alternative code optimization goals such as energy consumption. The algorithms are implemented in our retargetable code generation framework called OPTIMIST, where the target processor architecture is modeled in our architecture description language xADML.

1 Scientific project results 2001–2007

The integrated code generation problem is a combination of several NP-hard combinatorial optimization problems; assuming $P \neq NP$, an exact algorithm that generally finds an optimal solution in polynomial time can thus not exist. However, our ambition is to delay the combinatorial explosion of optimization time by suitable algorithmic techniques such that an optimal solution can be computed for practically relevant problem sizes within reasonable time. Faster heuristics for large problem instances are also considered.

Our first approach to optimal code generation was based on dynamic programming. The space of partial solutions is structured as a multidimensional grid, and domain-specific properties are exploited for pruning. Partial solutions are constructed in increasing order of the topmost optimization concern, which may be register need, execution time, energy consumption, or some linear combination of these.

In spring 2001 we extended our original dynamic programming approach by generalized instruction selection, taking *mutations* (equivalent sequences of instructions) into account for computing an optimal schedule. For the optimization of execution time on VLIW and superscalar processors, we introduced the new concept of *time profiles* [C19], which are used to additionally classify subsolutions. The time profile of a schedule contains only the information that may still be relevant for the scheduling of a subsequent instruction. This idea allows to apply a similar pruning strategy as for the space optimization.

We implemented this algorithm, using LCC as frontend and LEDA (later replaced by BOOST) as a platform for basic data structures. The results were very encouraging, as they show that our method is practical

for quite large basic blocks with up to 50 IR operations, which covers nearly all basic blocks occurring in practice [C19]. The implemented system is since then called OPTIMIST.

We have set up an initial version of an XML based hardware description language called ADML (Architecture Description Markup Language) [Lic1]. A parser for this language has been implemented and connected to OPTIMIST. ADML was later extended to xADML [X2,PhD1]. A graphical editor for xADML was prototyped 2006 in a bachelor thesis project [X4].

In [C20] we generalized our method to cover DSP architectures with irregular register sets and non-homogeneous memory structure. We invented the concept of a *space profile* which denotes for a given scheduling situation the set of kinds of locations (*residences*) such as register classes, memory modules etc., where a value currently resides. The various register classes form a lattice, based on the relative *versatility* of the registers of different register classes. These relations can be easily derived from a given instruction set and can be utilized in order to consider, at each scheduling situation, only those alternatives for residences that may be useful in future scheduling decisions.

We then discovered that some alternatives need not be considered because of certain local symmetry properties in many scheduling situations. We exploited this effect by pruning such "duplicates" where an equivalent alternative has already been considered. We could empirically show for computational cores of typical DSP applications that such local symmetries occur quite frequently. In spite of this, it turned out that the benefit of exploiting symmetry is on the average not at all that large as intuition suggested. In some cases the overhead of symmetry testing even outweighed the achieved reduction of the solution space. (Note that finding local symmetries in DAGs is NP-complete as well.) Dramatic (i.e., exponential order) reductions in the size of the solution space could only be achieved for very few cases [C23].

We adopted the quite detailed and generic model by Steinke et al. (2001) for energy consumption in processors, and developed methods for energy optimization and energy-constrained time optimization on the basis of our dynamic programming approach [I11].

Our testbed platform used for the experimental evaluation of our ideas has converged into a complete retargetable compiler system called OPTIMIST (www.ida.liu.se/∼chrke/optimist). This took quite some time but was important to demonstrate the approach, especially to our industrial project partners. In 2005 we made a first public release of the prototype, under GPL licence [S3]. Moreover, a translator from the Open Research Compiler's intermediate representation to OPTIMIST has been added, which provides further frontends for C++ and Fortran, and allows, at some degree, to exploit ORC's high-level program analyses and transformations [X1].

In 2005/06 we provided an alternative approach to optimal integrated code generation for regular embedded processors and VLIW architectures that is based on *integer linear programming* [I13]. Quantitative comparisons with dynamic programming show that the two approaches complement each other fairly well [C28].

Moreover, we analyzed fundamental properties of the solution spaces for the various scheduling methods occurring in our work and in the literature, and derived a hierarchy of schedules for VLIW architectures [I12]. This more theoretical work was extended in 2006 by an in-depth analysis of necessary properties of reservation tables in VLIW processor instruction sets that are required to guarantee that the dynamic programming algorithm produces an optimal schedule. A central result is our proof that scheduling anomalies are impossible if only instructions with *multi-block* reservation tables occur (and thus, greedy code compaction suffices in optimization), whereas scheduling anomalies can occur, with any scheduling method based on topological sorting, in the presence of just a single instruction with a non-multiblock reservation table [J10].

Since 2007, the project focus was on developing a genetic programming heuristic for OPTIMIST (submitted) and on ILP modeling of integrated local code generation and integrated software pipelining for clustered VLIW architectures. The project continues; 2006–2008 it is funded by VR.

2 Promotions, Degrees, Examina

- **Christoph Kessler** Habilitation 2001, Docent 2002, Professor 2007
- **Andrzej Bednarski** Lic. thesis *A Dynamic Programming Approach to Optimal Retargetable Code*

Generation for Irregular Architectures, Linköping studies in science and technology Thesis No. 1001, 2003;
Ph.D. thesis *Optimal Integrated Code Generation for Digital Signal Processors*, Linköping Inst. of Technology (LiTH), June 2006.

- **Mikhail Chalabine** Lic. thesis *Interactive Invasive Parallelization*, LiTH, Dec. 2007

3 Master thesis students

In total, I examined 24 master thesis projects 2001–2007. Here I only list those internal thesis projects that directly contributed to this Ceniit project.

- **Anders Edqvist**: *High-level optimizations for OPTIMIST.* LITH-IDA-EX–04/078–SE, Linköpings universitet, Oct. 2004.
- **22vid Landén**: *ARM9E processor specification for OPTIMIST.* LITH-IDA-EX–05/022–SE, Linköpings universitet, Feb. 2005. In cooperation with IAR Systems AB, Uppsala.
- **Yongyi Yuan**: *Optimization of amplifier code for the Motorola DSP 56367 processor with OPTIMIST*. LITH-IDA-EX–06/032–SE, Linköping university, April 2006. In cooperation with Softube AB, Linköping.
- **Andreas Rehnströmer**: *Xe: A graphical editor for writing xADML processor specifications.* LITH-IDA-EX-ING–06/006–SE, Linköping university - Institute of Technology, May 2006

4 Persons supported by the Ceniit grant 2001–2007

The following researchers were partly supported by the Ceniit grant:

- Christoph Kessler (project leader),
- Andrzej Bednarski (PhD student),
- Mikhail Chalabine (PhD student),
- Mattias Eriksson (PhD student).

Also, a small share of the salary cost of the PELAB secretary, Bodil Mattsson, was paid from the Ceniit grant, in accordance with PELABs internal policy. Finally, minor amounts were used to cover conference travel costs and to buy some computers and other necessary technical equipment.

5 Cooperation with industry

Since 2001 we cooperate with IAR Systems AB, Uppsala (contact person: Petter Edman) on code generation for DSPs. IAR provided valuable information and feedback to our work. We jointly supervised an exjobb project on porting OPTIMIST to the processors ARM9E and ARM9E-Thumb, which are important platforms for IAR and their customers [X2]. Moreover, Carl von Platen, who was our contact person at IAR until Jan. 2005, was the opponent in Andrzej Bednarskis licentiate defense in 2003. Carl joined Ericsson Research in Lund in 2005 and we continued cooperation on the integrated software pipelining problem.

2002-2004 we also cooperated with Freehand DSP (now VIA Technologies) in the form of meetings, presentations and common research proposals. This cooperation was not continued after our contact person at VIA, Dake Liu, left the company in 2005.

In another master thesis project we cooperated with Softube AB, Linköping (contact person: Oscar Öberg), on modeling the Motorola MC56K DSP processor as a target for OPTIMIST and quantitatively evaluating our system for this platform [X3].

We cooperated in several external master thesis projects on code optimization with Ericsson AB Softlab, most recently in projects on instruction cache optimizations, predication of conditional code, and speculative precomputation in just-in-time compilation for Ericsson's AXE/APZ platform.

6 Cooperation with other CENIIT projects

Close contacts exist to the computer architecture research group by Dake Liu (former Ceniit project 99.2). One line of cooperation discussed the possibility of coupling their DSP processor design tools and optimal code generation for a fixed DSP application. Placed in an optimization loop, this would eventually allow to automatize larger parts of the DSP processor design optimization process. To make this work, our hardware description language needs to be extended so that it is general enough to be used for this purpose.

Moreover, we provided (within very short time) a working compiler for the reconfigurable base-band processor BBP1 developed in Dake Liu's group, implemented by Anders Edqvist who had done his thesis project on IR transformations for OPTIMIST before [X1]. Further cooperations are planned.

We also cooperated with Uwe Assmann (former CENIIT project 01.03 on software composition) on developing composition-based strategies for source-program-level parallelization.

With Peter Bunus (CENIIT project 05.02 on high-level debugging techniques) we cooperated on *automatic roundtrip software engineering (ARE)* in aspect-weaving systems [C29].

7 Creation of a new research group

With the help of the Ceniit grant I could establish a research group on compiler technology, focussing on parallelization and code generation, with currently 2 PhD students as a permanent subgroup within the Programming Environments Laboratory (PELAB) at the computer science department (IDA). Currently my group is the second largest subgroup of PELAB.

8 List of publications 2001–2007

This is a list of my group's research-related publications 2001–2007, ordered by publication time.

[C19] Christoph W. Kessler, Andrzej Bednarski. A Dynamic Programming Approach to Optimal Integrated Code Generation. Proc. ACM SIGPLAN Workshop on Languages, Compilers, and Tools for Embedded Systems (LCTES'2001), June 22-23, 2001, Snowbird, Utah, USA.

[I9] Christoph W. Kessler. Managing Irregular Remote Accesses to Distributed Shared Arrays in a Bulk-Synchronous Parallel Programming Environment. Proc. of CPC'01 9th Int. Workshop on Compilers for Parallel Computers, Edinburgh (Scotland), pp. 195-204, June 2001.

[J7] Christoph W. Kessler. *NestStep*: Nested Parallelism and Virtual Shared Memory for the BSP model. *The Journal of Supercomputing* **17**(3), 2001.

[C20] Christoph W. Kessler, Andrzej Bednarski. Optimal Integrated Code Generation for Clustered VLIW Architectures. Proc. ACM SIGPLAN conference on Languages, Compilers, and Tools for Embedded Systems (LCTES-SCOPES'2002), June 19-21, 2002, Berlin, Germany.

Appeared also in *ACM SIGPLAN Notices* **37**(7), July 2002.

[I10] Andrzej Bednarski, Christoph W. Keßler. Optimal integrated code generation for VLIW architectures. Proc. CPC'03 10th Int. Workshop on Compilers for Parallel Computers, Amsterdam (The Netherlands), pp. 71-80, Jan. 2003.

[Lic1] Andrzej Bednarski. *A Dynamic Programming Approach to Optimal Retargetable Code Generation for Irregular Architectures*. Licentiate thesis, Linköping studies in science and technology Thesis No. 1001, Jan. 2003.

[C21] Mikhail Chalabine, Christoph W. Keßler, Staffan Wiklund. Optimising Intensive Interprocess Communication in a Parallelised Telecommunication Traffic Simulator. Proc. HPC-2003 High-Performance Computing (track of the Advanced Simulation Technology Conference), Orlando, Florida, USA, April 2003.

[J8] Christoph W. Keßler. Managing distributed shared arrays in a bulk-synchronous parallel environment. *Concurrency and Computation – Practice and Experience* **16**:133–153, 2004.

[C22] Christoph W. Keßler. A practical access to the theory of parallel algorithms. Proc. ACM SIGCSE'04 Symposium on Computer Science Education, Norfolk, Virginia, USA, March 2004.

[C23] Andrzej Bednarski and Christoph W. Keßler. Exploiting Symmetries for Optimal Integrated Code Generation. Proc. Int. Conf. on Embedded Systems and Applications (ESA'04), June 21-24, 2004, Las Vegas, Nevada, USA.

[I11] Andrzej Bednarski, Christoph Kessler. Energy-Optimal Integrated VLIW Code Generation. Proc. of CPC'04 11th Int. Workshop on Compilers for Parallel Computers, Seeon, Germany, pp. 227-238, July 2004.

[X1] Anders Edqvist. *High-level optimizations for OPTIMIST.* Master's thesis, LITH-IDA-EX–04/078–SE, Linköpings universitet, Oct. 2004.

[X2] David Landén. *ARM9E Processor Specification for OPTIMIST.* Master's thesis, LITH-IDA-EX–05/022–SE, Linköpings universitet, Feb. 2005.

[E2] Erik Altman, Jim Dehnert, Christoph W. Keßler, Jens Knoop [eds.]. *Scheduling for Parallel Architectures: Theory, Applications, Challenges.* Dagstuhl Seminar Electronic Proceedings, Dagstuhl Seminar 05101, www.dagstuhl.de, 2005.

[C24] Håkan Mattsson, Christoph Kessler. Towards a virtual shared memory programming environment for grids. Proc. PARA'04 Workshop on state-of-the-art in scientific computing, Copenhagen, June 2004. Springer LNCS, 2006.

[J9] Christoph W. Keßler, Andrzej Bednarski. Optimal integrated code generation for VLIW architectures. *Concurrency and Computation – Pract. Exp.* **18**:1353–1390, 2006.

[C25] Mikhail Chalabine and Christoph Kessler. Parallelisation of Sequential Programs by Invasive Composition and Aspect Weaving. Proc. Sixth Int. Workshop on Advanced Parallel Processing Technologies (APPT 2005), Hong Kong, China, 27-28 Oct. 2005, Springer LNCS.

[C26] Mikhail Chalabine and Christoph Kessler. *Identifying Crosscutting Concerns in Parallel Programs.* Proc. Hawaii Int. Conference on System Sciences (HICSS-39), Kauai, Hawaii, USA, Jan. 2006. IEEE.

[I12] C.W. Kessler, A. Bednarski. Classification and generation of schedules for VLIW processors. Proc. of CPC'06 12th Int. Workshop on Compilers for Parallel Computers, A Coruna, Spain, Jan. 2006, pp. 60-72.

[I13] A. Bednarski, C.W. Kessler. Integer Linear Programming versus Dynamic Programming for Optimal Integrated VLIW Code Generation. Proc. of CPC'06 12th Int. Workshop on Compilers for Parallel Computers, A Coruna, Spain, Jan. 2006, pp. 73-85.

[C27] Mattias Eriksson, Christoph Kessler, and Mikhail Chalabine. Load Balancing of Irregular Parallel Divide-and-Conquer Algorithms in Group-SPMD Programming Environments. Proc. 8th Workshop on Parallel Systems and Algorithms (PASA 2006), Frankfurt am Main, Germany, March 2006. *GI Lecture Notes in Informatics (LNI)*, vol. P-81, pp. 313-322, 2006.

[X3] Yuan Yongyi. *Optimization of amplifier code for the Motorola DSP 56367 processor with OPTIMIST.* Master thesis LITH-IDA-EX–06/032–SE, April 2006.

[X4] Andreas Rehnströmer. *A graphical editor for writing xADML processor specifications.* Master thesis LITH-IDA-EX-ING–06/006–SE, May 2006.

[PhD1] Andrzej Bednarski. *Optimal Integrated Code Generation for Digital Signal Processors.* Ph.D. thesis No. 1021, Linköping Studies in Science and Technology, Linköping, June 2006.

[I14] C. Kessler, P. Fritzson. NestStepModelica - Mathematical Modeling and Bulk-Synchronous Parallel Simulation. *PARA'06 State-of-the-Art in Scientific and Parallel Computing, Umea, Sweden, June 2006.*

[C28] Andrzej Bednarski, Christoph Kessler. Optimal Integrated VLIW Code Generation with Integer Linear Programming. *Proc. Euro-Par 2006*, Springer LNCS 4128, August 2006.

[C29] Mikhail Chalabine, Christoph Kessler, Peter Bunus. Automatic Round-trip Software Engineering in Aspect Weaving Systems. *Proc. 21st IEEE/ACM International Conference on Automated Software Engineering (ASE2006)*, Tokyo, Japan, Sept. 2006.

[DS1] Mikhail Chalabine: Invasive Interactive Parallelization, *Proc. of the Doctorial Symposium of the Fourteenth ACM SIGSOFT Symposium on Foundations of Software Engineering (FSE 14)*, Portland, Oregon, USA, November 2006.

[C30] Christoph Kessler, Peter Fritzson, Mattias Eriksson. NestStepModelica - Mathematical Modeling and Bulk-Synchronous Parallel Simulation. In: Bo Kågström, Erik Elmroth, Jack Dongarra, Jerzy Wasniewski (eds.): *PARA'06 State-of-the-Art in Scientific and Parallel Computing, Umea, Sweden, June 2006.* Springer LNCS 4699:1006–1015 (2007).

[C31] Mikhail Chalabine, Christoph Kessler: A Formal Framework for Automated Round-trip Software Engineering in Static Aspect Weaving and Transformations. Proc. ACM SIGSOFT/IEEE 29th Int. Conference on Software Engineering (ICSE-2007), Minneapolis, USA, May 2007.

[C32] Mikhail Chalabine, Christoph Kessler: A Survey of Reasoning in Parallelization. *Proc. 8th ACIS Int. Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2007)*, Qingdao, China, July 2007. IEEE.

[W1] Daniel Johansson, Mattias Eriksson, Christoph Kessler. Bulk-synchronous parallel computing on the CELL processor. *PARS'07: 21. PARS - Workshop*, Hamburg, Germany, May 31-Jun 1, 2007. GI/ITG-Fachgruppe Parallel-Algorithmen, -Rechnerstrukturen und -Systemsoftware (PARS). *PARS-Mitteilungen* 24, Dec. 2007.

[W2] Bert Wesarg, Holger Blaar, Jörg Keller, Christoph Kessler. Emulating a PRAM on a Parallel Computer. PARS'07: 21. PARS - Workshop, Hamburg, Germany, May 31-Jun 1, 2007. GI/ITG-Fachgruppe Parallel-Algorithmen, -Rechnerstrukturen und -Systemsoftware (PARS). PARS-Mitteilungen 24, Dec. 2007.

[J10] Christoph W. Keßler, Andrzej Bednarski, Mattias Eriksson. Classification and generation of schedules for VLIW processors. *Concurrency and Computation – Pract. Exp.* **19**:2369–2389 (2007).

[Lic2] Mikhail Chalabine. *Interactive Invasive Parallelization.* Licentiate thesis, Linköping studies in science and technology Thesis No. 1339, Linköping university, Dec. 2007.

[C33] Christoph Kessler, Welf Löwe. A Framework for Performance-Aware Composition of Explicitly Parallel Components. In C. Bischof et al. (eds.): *Parallel Computing: Architectures, Algorithms and Applications, Proc. ParCo-2007 conference, Jülich/Aachen, Germany, Sept. 2007*, IOS Press, Jan. 2008.